

DECnet-DOS

User's Guide

November 1989

This manual describes how to set up and use DECnet-DOS. It details the DECnet-DOS commands used for performing network tasks.

Supersession/Update Information: This is a new manual.

Operating System and Version: MS-DOS V3.1
MS-DOS V3.2
MS-DOS V3.3
MS-DOS V4.0

Software Version: DECnet-DOS V3.0

digital™

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital or its affiliated companies.

Copyright © 1989 by Digital Equipment Corporation
All Rights Reserved.
Printed in U.S.A.

The postage-prepaid Reader's Comments form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DECnet-DOS	VAX
DECmate	DECUS	VAXcluster
DECnet	ThinWire	VMS
digital ™	UNIBUS	VT

IBM is a registered trademark of International Business Machines Corporation.
PC/XT and Personal Computer AT are trademarks of International Business Machines Corporation.
MS, MS-DOS, and MS OS/2 are registered trademarks of Microsoft Corporation.
3Com is a trademark of 3Com Corporation.
MICOM-InterLan is a registered trademark of MICOM Systems, Inc.

Contents

Preface

1 Overview of DECnet-DOS

1.1	Introducing DECnet-DOS	1-1
1.1.1	DECnet Network Terms	1-3
1.2	DECnet-DOS Capabilities	1-4
1.3	DECnet-DOS Utilities	1-5

2 Using the DECnet-DOS Job Spawner

2.1	Overview of the Job Spawner	2-1
2.2	Creating a Job Spawner Database File	2-2
2.3	Using the Job Spawner	2-4

3 Using FAL to Provide Remote File Access to Your Node

3.1	Starting FAL	3-1
3.2	Entering FAL as an Object	3-2
3.3	File Access	3-3
3.4	Performance and Tuning Hints	3-4
3.5	The FAL Command Syntax	3-5

4 Using NFT to Access Local and Remote Files

4.1	Running the NFT Utility	4-2
4.2	Entering NFT Commands	4-3
4.2.1	Command Prompts	4-4
4.2.2	Command Switches	4-4
4.3	Using the HELP Command	4-5
4.4	Exiting NFT	4-6
4.5	Defining Remote Node Access Information	4-6
4.5.1	Defining Access-Control Information with NFT	4-7
4.5.2	Using Default Access-Control Information	4-8
4.5.3	Changing the Default Access-Control Information	4-9
4.5.4	Displaying Access-Control Information	4-10
4.6	File Specifications	4-10
4.7	Types of Files and Systems	4-11
4.8	File Operations	4-13
4.8.1	Appending Files	4-14
4.8.2	Copying Files	4-14
4.8.3	Deleting Files	4-15
4.8.4	Displaying Directory Information	4-16
4.8.5	Printing Files	4-16
4.8.6	Displaying a File's Contents	4-17
4.8.7	Running Command Files	4-17
4.8.8	Logging File Operations	4-18
4.9	NFT Command Summary	4-18
	APPEND	4-20
	COPY	4-23
	DELETE	4-29
	DIRECTORY	4-30
	EXIT	4-33
	HELP	4-34
	PRINT	4-36
	SET	4-37
	SHOW	4-38
	SUBMIT	4-39
	TYPE	4-40

5 Using NDU to Access Remote Printers and Disks

5.1	Installing the Virtual Device Drivers	5-5
5.2	Running NDU	5-5
5.3	Entering NDU Commands	5-7
5.3.1	Command Verbs	5-7
5.3.2	Entities	5-8
5.4	Using the HELP Command	5-11
5.5	Special Considerations While Using Virtual Disks	5-11
	CLOSE	5-13
	CREATE	5-15
	DELETE	5-17
	HELP	5-18
	OPEN	5-20
	SHOW	5-22

6 Using DECnet-DOS Mail

6.1	Running the Mail Utility	6-1
6.1.1	REPLY_ADDRESS	6-2
6.1.2	DEFAULT_NODE	6-3
6.1.3	CARBON_COPY	6-4
6.1.4	EDITOR	6-4
6.1.5	PERSONAL_MESSAGE	6-4
6.2	Starting the Mail Utility	6-5
6.2.1	Entering Mail Commands from the Command Line	6-5
6.2.2	Entering Mail Commands from the Mail Prompt	6-6
6.3	Using Distribution Lists	6-6
6.4	Using Qualifiers	6-8
6.5	Getting Help for Mail	6-9
6.6	Mail Command Summary	6-9
	EXIT	6-10
	HELP	6-11
	MAIL AND SEND	6-12
	SHOW DEFAULTS	6-14

A File Specifications for Accessing Remote Files

A.1	Specifying Remote Files	A-1
A.2	Using Wildcards	A-2

B NFT Error Messages

B.1	Types of NFT Error Messages	B-2
B.2	Command Line Error Messages	B-3
B.3	Warnings	B-5
B.4	File I/O Error Messages	B-6
B.5	Network Error Messages	B-7
B.6	Secondary Error Messages	B-8
B.7	Other Error Messages	B-10
B.7.1	Warnings	B-17

C NDU Error Messages

C.1	Virtual Disk Error Messages	C-2
C.1.1	Command Line Error Messages	C-2
C.1.2	Network Errors	C-3
C.1.3	Execution Error Messages	C-5
C.2	Virtual Printer Error Messages	C-9
C.2.1	Command Line Error Messages	C-9
C.2.2	Network Errors	C-10
C.2.3	Execution Error Messages	C-11

D Using FAL in a Windows Environment

D.1	Starting FAL	D-1
D.1.1	FAL Menu Options	D-2
D.1.1.1	Options	D-2
D.1.2	Help	D-3

E Using NFT in a Windows Environment

E.1	Starting NFT	E-1
E.2	NFT Menu Options	E-1
E.2.1	Copy	E-2
E.2.2	View	E-3
E.2.3	Change Directory	E-3
E.2.4	Special	E-4
E.2.5	Data Type	E-4
E.2.6	Character Set	E-5
E.2.7	Show Defaults	E-6
E.2.8	Help	E-6
E.2.9	Cancel (^C)	E-6

F Running Asynchronous DECnet Communications

F.1	Associated Documents	F-1
F.2	Running Asynchronous DECnet	F-2
F.3	Using the SETHOST Scripts	F-3
F.3.1	Script Commands	F-4
F.4	Using a Hardwired Dedicated DECnet Line	F-10
F.5	Using a Hardwired Dynamically Switched Terminal Line	F-10
F.5.1	The HARDDYN.SCR Script	F-10
F.6	Using a Modem with a Dedicated DECnet Line	F-12
F.6.1	The DIALNET.SCR Script	F-13
F.7	Using a Modem with a Dynamically Switched Terminal Line	F-15
F.7.1	The DIALDYN.SCR Script	F-15

G Modem Control

G.1	Using Modem Control	G-1
G.2	Modem States	G-2
G.2.1	General	G-2
G.2.2	State 1	G-2
G.2.3	State 2	G-2

G.2.4	State 3	G-2
G.2.5	State 4	G-4
G.2.6	State 5	G-4
G.2.7	State 6	G-5
G.2.8	State 7	G-5
G.3	Interface Requirements	G-5
G.4	Modem Option Selections	G-7

Glossary

Index

Figures

1-1	A Sample DECnet Network	1-2
4-1	Command Example	4-3
4-2	Copying a Local File to a Remote Node	4-11
5-1	Using Virtual Disks on DECnet-DOS	5-2
5-2	Using Remote Virtual Printers on DECnet-DOS	5-3
5-3	Parts of an NDU Command	5-7
G-1	Modem States	G-3
G-2	Minimum Implementation of V.24 Circuits for Full Duplex Modem Control	G-5

Tables

1-1	DECnet-DOS Utility Classifications	1-7
4-1	Copying Files from a Remote Node	4-12
4-2	Copying Files to a Remote Node	4-13
4-3	NFT Commands	4-19
5-1	NDU Commands for Virtual Disks	5-7
5-2	NDU Command Entities	5-8
A-1	Operating Systems and File Specifications	A-1
G-1	Asynchronous Mode, Direct Connection	G-6

G-2	Asynchronous Mode, Modem Connection	G-7
G-3	User-Selectable Modem Options	G-8

Preface

The DECnet-DOS™ software communication product enables individual computer systems, such as your personal computer, to communicate with one another in a network.

The *DECnet-DOS User's Guide* provides the information you need to operate a DECnet-DOS system within a DECnet™ network. This guide defines the basic DECnet network utilities, explains the terms and concepts related to DECnet networks, and defines and explains the command set for each basic utility of DECnet-DOS Version 3.0.

In this manual:

- The term "DOS" refers to MS-DOS™ or PC DOS™ running on supported personal computers.
- The term "DECnet-DOS" refers to DECnet-DOS Version 3.0 running on supported industry-compatible systems that run the DOS operating system.
- The term "personal computer" refers to all industry-compatible personal computers.

For a list of all supported personal computers and supported versions of the DOS operating system, see the *DECnet PCSA Client for DOS Software Product Description*.

Objectives of This Manual

The *DECnet-DOS User's Guide* describes the DECnet utilities (Job Spawner, File Access Listener, Network Device Utility, Network File Transfer, and Mail). An overview section describes DECnet network concepts and defines DECnet network terms. The network management utilities and terminal emulation utilities are included in separate guides. However, this guide includes an overview of networking terms, network management concepts, and terminal emulation.

You should be familiar with personal computers and the DOS operating system.

DECnet-DOS Getting Started provides an overview of the entire DECnet-DOS documentation set so that you can determine which books contain the information you need. Please read *DECnet-DOS Getting Started* before using the *DECnet-DOS User's Guide*.

Intended Audience

This guide is intended for users who want to expand the capabilities of their personal computers to share data and resources with other DECnet network systems. This guide explains the use of DECnet-DOS utilities, Network File Transfer (NFT), Network Device Utility (NDU), Mail (MAIL), File Access Listener (FAL), and the Job Spawner (SPAWNER).

How to Use This Manual

This manual consists of six chapters, seven appendixes, and a glossary:

- | | |
|-----------|--|
| Chapter 1 | Introduces the capabilities of DECnet-DOS. It also defines some basic concepts within a DECnet environment and introduces the DECnet-DOS utilities. |
| Chapter 2 | Describes how to use the DECnet-DOS Job Spawner to detect incoming connection requests from other nodes in the network. |
| Chapter 3 | Describes how to use the File Access Listener (FAL) utility for providing (or limiting) remote file access to your node from other nodes in the network. |

Chapter 4	Describes how to use the Network File Transfer (NFT) utility to access files on other nodes in the network.
Chapter 5	Describes how to use the Network Device Utility (NDU) to set up disk drives and use printers on remote nodes.
Chapter 6	Describes how to use the Mail utility to send and receive files between nodes on the DECnet network.
Appendix A	Describes file specifications in detail.
Appendix B	Lists the NFT error messages.
Appendix C	Lists the NDU error messages.
Appendix D	Describes the use of FAL in an MS-Windows environment.
Appendix E	Describes the use of NFT in an MS-Windows environment.
Appendix F	Explains how to run asynchronous DECnet communications.
Appendix G	Explains in detail the implementation of modem control.
Glossary	Contains a list of DECnet-DOS terms.

Graphic Conventions Used in This Document

The following graphic conventions are used in this manual:

Convention	Meaning
special type	Indicates examples of system output or user input. System output is in black; user input is in red.
COMMAND	Bold characters represent acceptable abbreviations for commands. For example, DELETE indicates that the acceptable abbreviation for the DELETE command is DEL .
UPPERCASE	In commands and examples, indicates that you should enter the exact characters shown. However, you can enter them in either uppercase or lowercase.
<i>italics</i>	In commands and examples, indicates a value that either the system supplies or you should supply.
{ }	Braces indicate that you are required to specify one (and only one) of the enclosed options. Do not type the braces when you enter the command.
[]	Brackets indicate that the enclosed data is optional. (If a vertical list of options is enclosed, you can specify only one option.) Do not type the brackets when you enter the command.
()	Parentheses enclose a set of options that must be specified together.
vertical list of options	A vertical list of options not enclosed within braces, brackets, or parentheses indicates that you can specify any number of options (or if defaults apply, none).
key	Indicates that you should press the specified key. CTRL/x indicates that you should hold down the Control key while you press the <i>x</i> key, where <i>x</i> is a letter.
Return	Indicates that you should press the key that executes commands. This key is Enter , Return , or ↵ , depending on your keyboard.

Associated Documents

The following documents are included in the DECnet-DOS documentation set:

- *DECnet-DOS Getting Started*

This manual provides an overview of DECnet-DOS. It also includes an explanation of the product set, a road map through the documents, a brief description of DECnet-DOS utilities and the tasks that you perform with them, and an introduction to frequently used DECnet-DOS commands.

- *Installing DECnet PCSA Client for DOS with Diskettes*

This guide describes procedures for installing and verifying the DECnet-DOS software on selected personal computers. This guide addresses a varied audience by providing the following:

- Instructions for using the automated installation procedure for the nontechnical end-user.
- Reference material for the technically oriented user who wants to bypass the automated system prompting.

- *DECnet-DOS SETHOST Terminal Emulation Guide*

This guide describes the SETHOST network virtual terminal utility. It explains how to use SETHOST and the set-up screens to connect to a host node and emulate a terminal connected to that node.

This guide also tells how to use scripts, which are text files containing commands that allow SETHOST to perform many operations automatically.

- *DECnet-DOS Network Management Guide*

This manual describes the Network Control Program (NCP), the Network Management Listener (NML), and the Loopback Mirror (MIRROR). It explains how to use NCP to manage your personal computer and monitor the local node; how to use NML to monitor remote nodes; and how to use loop commands and the MIRROR to perform loopback tests for troubleshooting. The management guide also details the commands and command syntax for NCP.

- *DECnet-DOS Programmer's Reference Manual*

This manual documents the programming interface and language library provided in the DECnet-DOS kit. This manual contains three parts: Part I provides a tutorial on writing network applications; Part II details the programming utilities and network programming calls

used in the creation of DECnet-DOS application programs; Part III provides reference information in six appendixes.

You should also have available the installation guide and manuals for the DOS operating system and for your personal computer.

It is important that you also read the contents of the file **README.TXT**, which is included on the first floppy diskette of your installation kit.

Overview of DECnet-DOS

Individual personal computers have a finite amount of space to store information. You can increase this capacity by using removable diskettes and back-up devices. However, information stored on these media is not quickly accessible, and you are still limited to using only the information entered into your system at your location.

Your personal computer can access additional information from remote locations when it is part of a network: two or more computers connected by communication lines (see Figure 1-1).

The *DECnet-DOS User's Guide* provides:

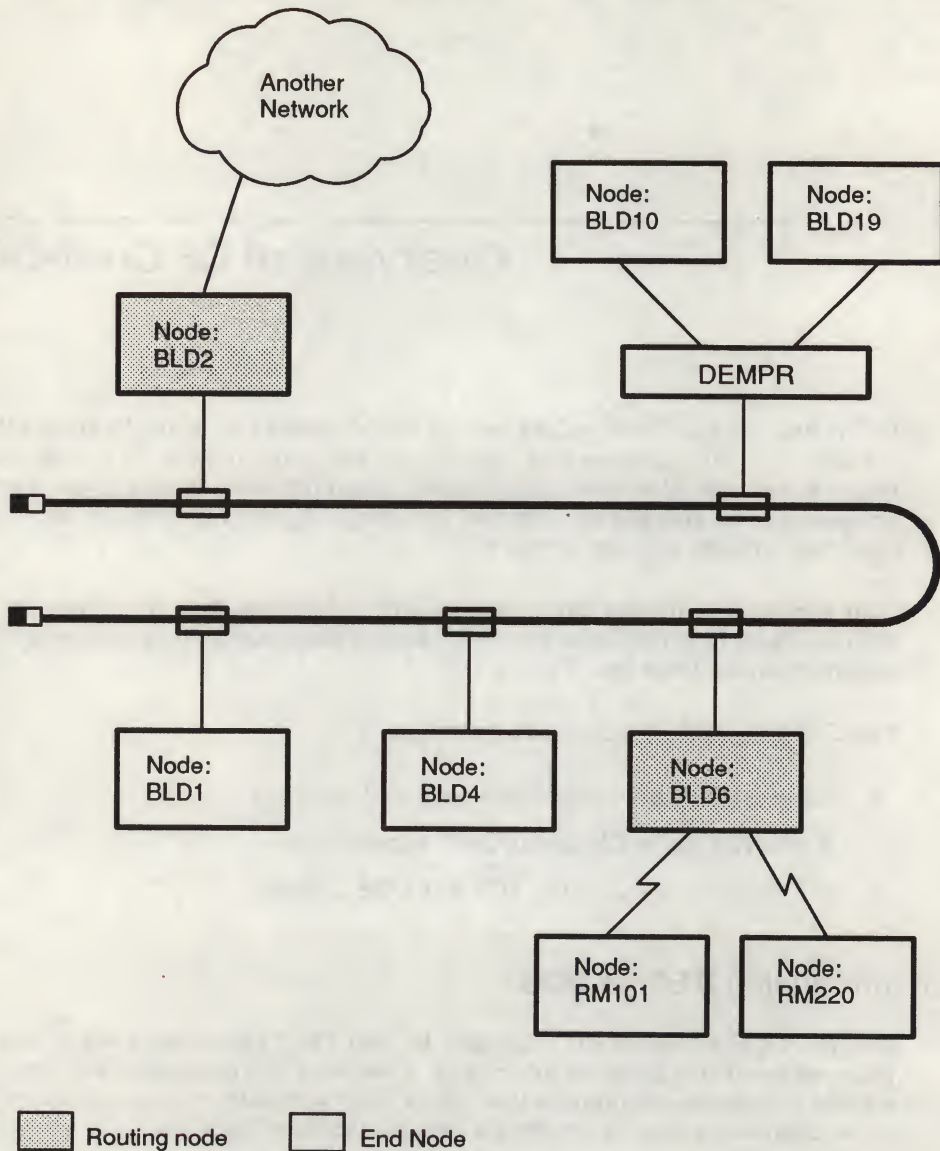
- An overview of networking terms and concepts
- A description of DECnet-DOS™ capabilities
- Information about using DECnet-DOS utilities

1.1 Introducing DECnet-DOS

DECnet-DOS software, an extension to your DOS operating system, enables your personal computer to interact in a network environment. DECnet-DOS software provides commands that allow your computer to communicate with other computers over the network communication lines.

Figure 1-1 illustrates a sample DECnet™ network.

Figure 1-1: A Sample DECnet Network



LKG-3162-891

As a user at a node on a DECnet network, you can:

- Send and receive mail
- Transfer files between your system and another on the network
- Print a file on a printer connected to another system on the network
- Access a file residing on a disk attached to another computer on the network as if it were a disk on your own personal computer

In a DECnet-DOS V3.0 environment, you can set up your node as an Ethernet Phase IV end node, by using an Ethernet communications controller and Ethernet hardware to connect to a Local Area Network (LAN).

1.1.1 DECnet Network Terms

The DECnet software allows connections to any Digital Equipment Corporation system, many personal computers, and, with the appropriate hardware and software configurations, to SNA networks and Packet Switched Data Networks (PSDNs). Each of these component systems on a DECnet network is called a **node**.

Throughout this guide, the term "node" refers to any processor, intelligent terminal or other computer system capable of functioning independently within a DECnet network environment. Figure 1-1 displays different types of nodes in the context of a DECnet network:

- An **adjacent node** is physically connected to another node by a single communications line, such as BLD1 and BLD4.
- An **end node**, such as BLD10, BLD19, BLD1, BLD4, RM101, or RM220, can only receive information for its own use; it cannot receive messages and subsequently route them to another node.
- A **routing node**, such as the shaded nodes in Figure 1-1 (BLD2 and BLD6), can receive and forward information from one node in the DECnet network to another node or network.
- An **executor node** can perform network management functions.
- A **local node** is the node on which you are physically working when you type in commands. For instance, if you are typing commands at the node RM101, RM101 would be your local node.
- A **remote node** is any node in the network other than your local node. For example, if your local node is RM101, all other nodes in the sample network in Figure 1-1 are remote nodes.

In a DECnet-DOS environment, you can set up your node as either:

1. An asynchronous DECnet Phase IV end node, using Digital Data Communications Message Protocol (DDCMP) over the asynchronous serial communications port.
2. An Phase IV end node, using an Ethernet communications controller and Ethernet hardware to connect to a LAN.

1.2 DECnet-DOS Capabilities

When you are using DECnet-DOS, you have the following network capabilities:

- **Remote file access**—Allows access to files on remote nodes. You can store and retrieve information on remote nodes. Remote file access allows file sharing among network nodes.
- **File transfer**—Allows exchange of files with other nodes.
- **Resource sharing**—Eliminates the need to duplicate resources at each node. With resource sharing, many nodes can use the same printers, storage facilities, and processing capabilities.
- **Virtual terminal support**—Allows you to use your personal computer as if it were a terminal directly connected to a remote host. Virtual terminal support is described in the *DECnet-DOS SETHOST Terminal Emulation Guide*.
- **Mail**—Allows you to send messages to remote nodes in the network.
- **Network management**—Provides an interface for controlling, monitoring, and testing DECnet-DOS software, as well as for configuring your node. Network management is described in detail in the *DECnet-DOS Network Management Guide*.
- **Task-to-task programming**—Allows you to write and use your own task-to-task programs. For information about programming for DECnet-DOS, refer to the *DECnet-DOS Programmer's Reference Manual*.

DECnet-DOS also provides various servers for use with the network utilities listed. These servers are dispatched by the Job Spawner when requested, or you can choose to run them as foreground tasks.

- **File Access Listener (FAL)**—Runs when the Job Spawner receives a file access request. Enables file access and transfers for such utilities as Network File Transfer (NFT) and Network Device Utility (NDU).
- **Loopback Mirror (MIRROR)**—Runs as a server when invoked by the NCP MIRROR command.

The Network Management Listener (NML) is a DECnet tool that listens for incoming network management connection requests from remote nodes and dispatches those commands on the local node.

1.3 DECnet-DOS Utilities

The following list summarizes the DECnet-DOS utilities used to support the capabilities listed in the previous section.

- **Job Spawner (SPAWNER)**

The Job Spawner allows your computer to act as a server for multiple service functions. The Job Spawner activates DECnet servers on your node.

You can also design and write your own server programs to be run by the Job Spawner. For more information about writing your own programs, refer to the *DECnet-DOS Programmer's Reference Manual*.

- **File Access Listener (FAL)**

FAL provides file access to your computer from other nodes in the network.

FAL is available as both a command line utility and an MS-Windows utility. This means that you can issue FAL commands from a command line, or you can use FAL in a windows environment. FAL functions the same way in both environments, but it has a very different appearance when you use the windows version. The windows version of FAL is discussed in Appendix D of this guide.

- **Network File Transfer (NFT)**

NFT transfers files between your local node and remote nodes. It also provides other file transfer services.

NFT is available as both a command line utility and an MS-Windows utility. This means that you can issue NFT commands from a command line, or you can use NFT in a windows environment. The NFT commands and functions are essentially the same in both environments, but they have a very different appearance when you use the

windows version. The windows version of NFT is discussed in Appendix E of this guide.

- **Network Device Utility (NDU)**

NDU defines virtual disk drives and virtual printers on remote systems.

NDU capabilities include:

- Defining files on remote systems as virtual disks and using them as if they were directly connected to your computer. Defining these remote virtual disks lets you share files with other users in the network.
- Directing text files to a remote node to be queued for later printing.

- **Mail Utility (MAIL)**

The Mail utility lets you send Mail messages and text files to other users in the network.

- **SETHOST**

The SETHOST terminal emulation program, allows you to log on to a remote host node. SETHOST lets you use your personal computer or workstation as if it were directly connected to the host node, giving you access to the host's resources. DECnet-DOS SETHOST is discussed in the *DECnet-DOS SETHOST Terminal Emulation Guide*.

- **Network Control Program (NCP)**

NCP allows you to perform network management tasks and to test the network hardware and software. Some NCP testing is run as a part of the installation procedure before you can use your personal computer as a node in the DECnet network. Any node on the DECnet network can use the Loopback Mirror, when invoked by NCP, to verify their network connection.

Information is also provided about the network management server Network Management Listener (NML). Nodes can send NCP commands to remote nodes running NML, and NML executes those commands and sends the result back to the source node. NCP is discussed in detail in the *DECnet-DOS Network Management Guide*.

You can start any of the DECnet-DOS utilities from the MS-Windows environment. Two of the utilities operate as fully functional windows applications. Seven of the utilities operate as standard applications. They work the same as if you started them from the command line, but they appear as part of a window. Many standard applications can run windows, sharing the work area with other applications in their own windows. However, some standard applications require the entire screen. One utility operates as a standard application that requires the entire screen. When you exit from this application, you return to the window environment you were using previously.

Table 1-1 shows the window classifications and the utilities.

Table 1-1: DECnet-DOS Utility Classifications

Fully Functional Windows Applications	Standard Applications	Standard Applications (requiring the entire screen)
FAL NFT	DTR DTS SPAWNER MAIL NCP NDU DECnet Menus	SETHOST

The DECnet Menus utility is a step-by-step introduction to the DECnet-DOS utilities. This Menus utility is described in *DECnet-DOS Getting Started*.

Using the DECnet-DOS Job Spawner

The Job Spawner is a DECnet-DOS™ utility that runs as a foreground task while waiting for incoming connection requests from other nodes in the network. When a connection request arrives, the Job Spawner looks up the requested object name or number in its database (DECOBJ.DAT), then runs the program that services that type of request. When the program finishes, the Job Spawner resumes control and again waits for incoming connection requests.

This chapter covers the following topics:

- An overview of the Job Spawner (see Section 2.1).
- Creation of the Job Spawner database file (see Section 2.2).
- How to use the Job Spawner (see Section 2.3).

2.1 Overview of the Job Spawner

In DECnet-DOS Version 3.0, the Job Spawner supports the File Access Listener (FAL), the DECnet™ Test Receiver (DTR), and user-written servers. By using the Job Spawner, your personal computer can process different connection requests at different times without any need for user intervention.

As an example, if the Job Spawner is running and it detects a request for file transfer, the Job Spawner initiates the FAL utility and causes FAL to run. FAL continues to run until the requested activity is complete. The Job Spawner then continues to listen for other requests. (The Job Spawner cannot run two service programs simultaneously.)

The Job Spawner accesses a database file to determine the programs (or objects) that run for each kind of connection that the Job Spawner detects. DECOBJ.DAT is the object database used by the Job Spawner.

The DECOBJ.DAT database is created for you during the installation procedure. However, if you installed DECnet-DOS manually, you might not have created this database. You must use the Network Control Program (NCP) command, DEFINE OBJECT, to create the DECOBJ.DAT database. You can use the NCP DEFINE OBJECT, LIST OBJECT, and PURGE OBJECT commands to change, review, or delete the objects in your database file. Refer to the *DECnet-DOS Network Management Guide* for more information about using NCP commands.

2.2 Creating a Job Spawner Database File

To create a DECOBJ.DAT file, first start the NCP utility.

```
C:\>NCP Return
```

```
NCP>
```

To define the objects, use the DEFINE command, which has the following format:

```
DEFINE OBJECT object-name NUMBER number FILE file-name  
[ARGUMENTS xxx]
```

where

object-name

is the name of the DECnet object. The object name can have a maximum length of 16 characters.

number

is the number of the DECnet object. The range for this number is 0 to 255.

file-name

is the name of a command file or program to be started by the Spawner. Command file names must end with the .BAT extension. The file name can include a path specification.

xxx

specifies the command line arguments for the program specified with **FILE**. To store argument information in uppercase letters, type the characters. To store argument information exactly as typed, place the string in double quotation marks. For example:

abc is stored in the database as **ABC**.

"Aab" is stored in the database as **Aab**.

Digital Equipment Corporation reserves the numbers 1–127 for Digital-supplied objects. Each number is predefined for a network program. The object numbers 128–255 are available for user-written or user-supplied applications. The following table lists the DECnet-DOS predefined object numbers:

Object #	Object Name	Object File Name
0	Any object	Any object file name
17	FAL	FAL.EXE
63	DTR	DTR.EXE
128-255	Any user-written/ user-provided network object	Any user-defined name

For example, to define FAL as object 17 with an argument that includes the command file T.CMD, use the following command:

```
NCP>DEFINE OBJECT FAL NUMBER 17 FILE FAL.EXE ARGUMENT "/LOG:T.CMD /ASCII" Return
```

Now you can use the **LIST** command to display the objects you have defined. For example:

```
NCP>LIST KNOWN OBJECTS Return
```

Permanent DECnet Object database as of 21-Apr-1989 13:24:21

Object Number Use File

```
FAL          17  1      C:\DNETDOS\FAL.EXE  "/LOG:T.CMD /ASCII"  
NCP>
```

You can also use the **PURGE OBJECT** command to delete one object or all of the objects in the database. For more information about **PURGE** and other NCP commands, refer to the *DECnet-DOS Network Management Guide*.

The object number zero has a special meaning. An unlimited number of objects can be defined as object zero. When DECnet receives a connection request of object number zero, it makes the connection based on object name.

To write your own server programs, note that the server program is run by the Job Spawner as if the following command were typed:

PROGRAM -USE *n*

where

n is the number of a socket on which an incoming connect request has been accepted by the Job Spawner in deferred mode. The server program is responsible for accepting or rejecting incoming connection requests. The server program is also responsible for closing all sockets used for data communications. This includes the socket that was passed to the server program by the Job Spawner.

2.3 Using the Job Spawner

To start the Job Spawner, simply type SPAWNER at the system prompt and press **Return**. To exit from the Job Spawner, type an exclamation mark (!).

You can also start the Job Spawner from a window, if you are using MS-Windows. The Job Spawner runs under windows as a standard application. (There is a SPAWNER.PIF file included in your kit that allows you to run the Job Spawner as a standard application.)

Format

SPAWNER [/LOG
 /HELP]

where

/LOG Logs messages about Spawner and incoming connections to the file SPAWNER.LOG in the DECnet directory.

/HELP Prints help information on the screen.

The following example illustrates the Spawner running and receiving an incoming connect request for FAL:

```
C:\>SPAWNER Return
press '!' to abort
SPAWNER (V3.0) listening... on Wed Jul 15 1989 at 11:22:46
Connect request from node RM204 for object #17 name fal on Wed Jul 15 1989
at 11:26:57
Executing: FAL -use 2... on Wed Jul 15 1989 at 11:26:57
          FAL - File Access Listener - Version 3.0
Network Driver Version 3.0
Current working directory:C:\DECNET
Files will be sent as either ASCII or binary.
Existing files will be overwritten.
No access checking will be done (world has read/write privileges).
FAL running....
DIRECTORY access from RM204 for LOCAL""::C:\DECNET\*.EXE;*
SPAWNER (Version 3.0) listening... on Wed Jul 15 1989 at 11:27:01
SPAWNER exiting... on Wed Jul 15 1989 at 11:34:22
```

The command line argument for FAL in this example is "-use 2." This means that the incoming connect request is pending (in accept deferred mode), on socket number 2.

If you start the Job Spawner and it cannot locate the necessary database to use for answering a request, it displays the following help information:

Error: Could not open file C:\DECNET\DECOBJ.DAT

- . Abort job Spawner by pressing '!'
- . Job Spawner does not support multiple links.
- . Database file for Spawner in DECnet database path.
- . Job Spawner can spawn batch files.
- . Job Spawner can pass command line arguments to servers.
- . Name of database file is DECOBJ.DAT.
- . Use the DEFINE and LIST commands in NCP to configure DECOBJ.DAT.

Example contents of a DECOBJ.DAT:

DECnet Objects

#	Taskname	File	"Arguments"
0	SHOWME	C:\SHOWME.BAT	"a1 a2 a3"
17	FAL		"/l:fal.log /a"
63	DTR		
129		C:\BATCH.BAT	"a1 a2"

In this example, several batch files are included that are associated with object names or object numbers. The Job Spawner executes each batch file as specified, passing the arguments that are contained in the DECOBJ.DAT database. The Job Spawner always attempts to close the socket that was created for the batch file's incoming connect request.

If the Job Spawner is started running and a connection request arrives at the node, the following message will appear:

Connect request from node BLD3 for object #17 name FAL on Dec
16, 1989 at 13:12:49.

This example shows that a user on node BLD3 is running NFT. FAL is responsible for handling remote NFT requests, so the Job Spawner runs a FAL process for the request.

Using FAL to Provide Remote File Access to Your Node

This chapter describes the File Access Listener (FAL). The FAL utility provides a way for you to let other nodes access files on your node. The purpose of this utility is to listen for and receive remote access requests from the network. These requests are the results of file access routines from other nodes in the network (such as NFT on another personal computer node or a COPY command from a DECnet-VAX™ system).

This chapter covers the following topics:

- Starting FAL (see Section 3.1).
- Entering FAL as an object (see Section 3.2).
- File Access (see Section 3.3).
- Performance and tuning hints (see Section 3.4).
- The FAL syntax (see Section 3.5).

3.1 Starting FAL

FAL must be running on your system before any exchanges of file data can take place. In addition, because DOS is a single-tasking operating system, you must run FAL to the exclusion of all other application programs.

You can use FAL in three different ways: from the command line, spawned by the Job Spawner, or in a windows environment (if you are using MS-Windows). FAL runs as a fully functional windows application. Use the file FAL.EXE if you want to run FAL as a windows application. This image file contains both windows and nonwindows versions of FAL. If you are not running windows, use the file FALNOWIN.EXE to save user disk storage space.

The windows version of FAL provides the same functions as command line FAL, but the appearance is very different. This chapter describes how to use FAL from the command line. For information on how to use FAL as a windows application, see Appendix D.

To start FAL from the command line, type the command at the system prompt. For example:

```
C:\>FAL Return
```

FAL runs until you cancel it by pressing any key. If you use CTRL/C, FAL does not finish the current task. Instead, it exits immediately and close down all links with the remote node. If you use any other key, FAL first finishes its current task and then exits.

If FAL is run through the Job Spawner, the Job Spawner runs in the foreground, waiting for incoming connection requests. When a connection request is made for a FAL connection, the Job Spawner dispatches FAL. FAL returns control to the Job Spawner when its connection is completed.

3.2 Entering FAL as an Object

If you ran the DECnet™ PCSA Client for DOS installation procedure to install DECnet-DOS™, FAL is installed as an object in the DECOBJ.DAT database. However, if you did not select FAL as an object to install in the installation procedure or if you installed DECnet-DOS manually, you can install it as an object now.

To enter FAL as an object, invoke NCP and use the DEFINE OBJECT command.

Objects are run by the Job Spawner, as needed. The Job Spawner listens for connection requests from remote systems. For more information on the Spawner, refer to Chapter 2 .

Format

DEFINE OBJECT FAL NUMBER 17 FILE FAL.EXE [ARGUMENTS *xxx*]

where

xxx specifies the command-line arguments for the program specified by *file-name*. You can use one to three arguments for each file.

For example, to define FAL as object #17 with an argument that includes the command file T.cmd, use the following command:

```
C:\>NCP DEFINE OBJECT FAL NUMBER 17 FILE FAL.EXE ARGUMENT "/LOG:T.CMD /ASCII"
```

3.3 File Access

When FAL detects a request to copy a file to or from your node, it first determines whether access checking has been enabled on your local system. (Access checking is enabled if the file DECACC.DAT exists in the DECnet database path. The file is created the first time you use the NCP SET ACCESS command. For more information about using NCP and NCP commands, refer to the *DECnet-DOS Network Management Guide*.) If access checking is not enabled, your system's files are available to any and all incoming access requests while FAL is running.

NOTE

If you issue the NCP command PURGE KNOWN ACCESS, you clear the contents of the DECACC.DAT file. The DECACC.DAT file will no longer exist. **If this happens, FAL will not check access and will allow access to any user.**

If access checking is enabled, then FAL checks the type of access privilege contained in the incoming request. If the request has the proper privilege for access to your node, FAL sends or receives the file data to or from the requesting node.

NOTE

MS-DOS systems do not retain file attributes with files. As a result, when FAL is asked by a remote node for the attributes of a file on the personal computer, it cannot determine what those attributes are.

3.4 Performance and Tuning Hints

This section provides hints and suggestions for improving performance or to tune your system. Also listed are some restrictions you might encounter while using FAL on a DOS system.

- Some hosts might use as many as six logical links to perform a single file access function. If you plan to run FAL, you should increase the number of logical links supported. Enter the following NCP command:

```
NCP>DEFINE EXECUTOR MAXIMUM LINKS 6 Return
```

Refer to the *DECnet-DOS Network Management Guide* for more information on the DEFINE command.

The default MAXIMUM LINKS for the DEFINE EXECUTOR command is 4.

- FAL does not perform file locking. If you will be using FAL for simultaneous access to the same file, ensure that all users of that specific file open the file for reading **only**. If two users attempt to open the same file for simultaneous write, unpredictable results might occur.
- Wildcard file access is supported.
- VMS cannot use path specifications that are not surrounded by double quotation marks. Some restrictions apply when using file specifications enclosed in quotation marks with VMS. To solve all these problems, FAL accepts VMS syntax for file specifications. For example:

```
$ DIR MSDOS::"\TEST\*.C" Return will fail
```

```
$ DIR MSDOS::[TEST]*.C Return will work
```

- \whatever is sometimes a file and sometimes a directory. To avoid the problems that this causes, always specify file names or use VMS syntax:

```
$ DIR MSDOS::"\ *.*" Return will fail
```

```
$ DIR MSDOS::"\ *.*" Return will work
```

```
$ COPY *.whatever MSDOS::"\DIR\ *.*" Return will fail
```

```
$ COPY *.whatever MSDOS::[DIR]*.* Return will work
```

- VMS does not separate the parts of a file specification when it uses wildcards. You cannot make wildcard copies of files from a personal computer to VMS by using VMS DCL with DOS file syntax. Such copies must be done using VMS file syntax:

```
$ COPY MSDOS::[WORK]*.*.* Return will work
```


- For best performance, a USE-COUNT of at least 4 is recommended. This is the default value supplied by the DECnet PCSA Client for DOS installation procedure. You can have a use count as high as your number of MAXIMUM LINKS.

3.5 The FAL Command Syntax

While FAL is running, file data can be passed back and forth between your node and the requesting node. You can control the way FAL operates by using switches.

FORMAT

```
FAL [ /ASCII
      /BINARY
      /ERROR
      /LOG:file-name ]
```

where

/ASCII transfers all files as ASCII files. An ASCII file is a file whose records end with end-of-text or line-terminating characters, such as a carriage return/line feed (CR/LF) pair.

Image files can lose data if they are copied in the ASCII format.

/BINARY transfers all files as binary files. A binary (or image) file is a file whose data is copied exactly as it appears with no interpretation of CR/LF characters.

Using this switch ensures that any file copied to a remote node and then back to your personal computer will not lose any data. However, ASCII files copied from the personal computer when the /BINARY switch is used might not be in the proper format for use on the remote node.

/ERROR reports an error to the remote node if the requesting node is attempting to overwrite an existing file. If you do not set this switch, an existing file is deleted and a new one created with the same name. The default is NO ERROR.

/LOG:file-name logs the type of access for every request it receives. The type of access includes the following information:

- The command or request (such as an NFT DIRECTORY or DELETE command)
- The requested directory name and file name
- The node name
- The access-control information for that node (user name and account)

This information is displayed on the screen. If you do not want to see the information on the screen, you can redirect it to an output file when you first issue the FAL command.

file-name is the name of a file to use for storing the access information.

To use these switches, type them after the FAL command on the same line. Leave a space between the command and the switch. If you use more than one switch, be sure to separate them with a space.

NOTE

When FAL is started without either switch (/ASCII or BINARY), it automatically checks each file. FAL looks for any CR/LF characters in the first 512 bytes of the file. If FAL finds CR/LF character, it sends the file as ASCII.

Examples

The following command starts FAL, requesting it to report errors to the remote node that requested the transfer and then log all file transfer requests in the file LOGFILE.TXT.

```
C:\> FAL/ERROR/LOG:LOGFILE.TXT Return
```

The following command enters FAL as an object in DECOBJ.DAT.

```
C:\> DEFINE OBJECT FAL NUMBER 17 USE-COUNT 8 ARG"/ERROR/LOG:LOGFILE.TXT" Return
```

The Job Spawner runs this object if a connection request is made for a file transfer. When a remote node makes this request, FAL runs the same as in the previous example. The only difference is when the connection request completes, FAL exits until another remote node requests a file transfer.

Using NFT to Access Local and Remote Files

File Transfer is a basic utility of networks. Without it you could not easily share files with other nodes on the network.

The Network File Transfer (NFT) utility is a network program that allows you to access remote files. NFT provides access to files on any DECnet™ system that is running a File Access Listener (FAL). NFT also supports numerous file attributes.

The NFT utility allows you to:

- Append two or more files
- Copy files between the local and remote nodes
- Delete local and remote files
- List files located in a local or remote directory
- Define and display access-control information
- Run command files on remote nodes
- Display the contents of a local or remote file on your screen
- Print files on remote printers

This chapter describes:

- Running NFT (see Section 4.1).
- Entering NFT commands (see Section 4.2).
- Getting help (see Section 4.3).

- Exiting NFT (see Section 4.4).
- Defining remote node access information (see Section 4.5).
- File specifications (see Section 4.6).
- Types of files and systems (see Section 4.7).
- File operations (see Section 4.8).
- The NFT commands and their syntax (see Section 4.9).

You can use NFT in two different ways: from the command line, or in a windows environment (if you are using MS-Windows). NFT runs under MS-Windows as a fully functional windows application. Use the file `NFT.EXE` if you want to run NFT as a windows application. This file contains both windows and nonwindows versions of NFT. If you are not running windows and you want to save disk storage space, you should only use the file `NFTNOWIN.EXE`.

The windows version of NFT provides the same functions as command line NFT, but the appearance is very different. This chapter describes how to use NFT from the command line. For information on how to use NFT as a windows application, see Appendix E.

4.1 Running the NFT Utility

You perform NFT functions by using one of the following methods:

- Single command method:

```
C:\>NFT command 
C:\>
```

Note: When you run NFT in single command mode, an `ERRORLEVEL` of 0 is returned for success. A 1 is returned for failure.

- Single command method using a command file:

A command file, also called a script file, contains all the NFT commands that you want to use to complete a series of NFT operations. By using a command file, you need issue only one command to process all other commands.

To use a command file, you must enter a left angle bracket (<) followed by the file name. Then press ; for example:

```
C:\>NFT < COMMANDS.DAT 
C:\>
```

In this example, the COMMANDS.DAT file contains all of the NFT commands you want to process. The angle bracket is known as a redirect symbol.

- Multiple command method:

```
C:\>NFT 
NFT>command 
NFT>command 
NFT>EXIT 
C:\>
```

4.2 Entering NFT Commands

The NFT commands consist of three parts:

- The command verb and switch
- The source file specification
- The destination file specification

A file specification can be local or remote. If the file is local, you need not include the node name in the file specification. If the file is remote, you must include the node name in the file specification.

Figure 4-1: Command Example

NFT>COPY/PRINT FILE1.TXT RM10::FILE1.TXT;1 <input type="button" value="Return"/>		
Verb and Switch	Source File Specification	Destination File Specification

You must separate each part of the command with a space or tab. Notice in Figure 4-1 that the command verb and switch make up one part of the command line and so need not be separated. The node name is part of the file specification and should not be separated from the rest of the file specification.

You can abbreviate command verbs and switches to a character string, of three or more characters that is unique to that verb. For example, you can type **SHO** for **SHOW**. For clarity and consistency, all examples in this manual show the full command format.

If you need help with a command or switch, remember to use the **HELP** command (see Section 4.3).

4.2.1 Command Prompts

In many cases, if you type a command verb alone, NFT prompts you for further information. For example, to append one or more files to the end of an existing file if you type the **APPEND** command verb alone, NFT prompts you for the required information, as shown in this example:

```
NFT>APPEND 
File(s)? BLD3"SMITH OPEN": :WRIT: [SMITH] FILB.TXT;3 
To? FILA.TXT 
NFT>
```

4.2.2 Command Switches

Some operating systems store files with attributes that indicate the type of file and the format of the file's records. A DOS file does not include attributes. Therefore, when you copy a file from the local system to another system that stores attributes (such as VMS), you can use switches to indicate the type of file you are copying and the type of system it is coming from. NFT provides defaults for file attributes. You can use switches to change these defaults.

You can also use switches to modify command verbs. Switches are global in effect and must immediately follow the command verb in the command line. The same switch can have a different effect, depending on whether you are copying a file to or from a remote node.

A valid switch for several NFT commands is **/NOLOG**. This switch requests that NFT not print a notification line after the **APPEND**, **COPY**, **DELETE**, **PRINT**, and **SUBMIT** operations. It is ignored for all other commands.

If you use a switch that NFT does not accept, such as **/PRINT** with a remote input file, NFT displays an error message.

4.3 Using the HELP Command

If you need assistance in selecting NFT commands and switches, use the HELP command. Type:

```
NFT>HELP Return
```

The system responds with:

Help is available on the following commands:

APPEND	COPY	DELETE	DIRECTORY
EXIT	HELP	PRINT	SET
SHOW	SUBMIT	TYPE	

switches:

/ALLOCATION	/ASCII	/BLOCK	/BRIEF
/CC	/DELETE	/FIXED	/FULL
/IMAGE	/LSA	/MACY11	/MRS
/NOCONVERT	/NOLOG	/NOSPAN	/PRINT
/STREAM	/SUBMIT	/UNDEFINED	/VARIABLE /VFC

General command format is:
command/switches file file

Please note: Switches MUST always immediately follow the command.
Wildcards and file lists are supported for most commands.

```
NFT>
```

To obtain information about the SHOW command, for example, type:

```
NFT>HELP SHOW Return
```

SHOW

The SHOW command displays defaults for remote file access.

Example:

```
NFT> SHOW node::
```

```
NFT>
```

To obtain information about the /ASCII switch, type:

```
NFT>HELP /ASCII Return
```

/ASCII

The ASCII switch indicates that a file contains ASCII text.
If neither the ASCII nor the IMAGE switch is used, NFT attempts to determine the file type automatically.

This switch is valid only with the COPY and APPEND commands.

```
NFT>
```

4.4 Exiting NFT

You can use **CTRL/Z** **Return** to exit from NFT, or you can type **EXIT** followed by **Return**.

4.5 Defining Remote Node Access Information

Access-control information is security information that allows you to access a specified remote node with the privileges of a specific user. This information includes:

- **User name**

A character string consisting of 1 to 39 alphanumeric characters that identifies the user at the remote node.

- **Password**

A character string consisting of 1 to 39 alphanumeric characters that identifies the user's password.

- **Account**

A character string consisting of 1 to 39 alphanumeric characters that identifies the user's account.

You must follow the node name or node address (*area.node*) with the user name, password, and account enclosed in quotation marks (" "). Each field must be separated by a space. You must then follow the entire string with a double colon (::). The following is an example of a node name with access-control information:

```
BLD3"SMITH OPEN"::
```

For extra security, you can eliminate the password from the string and have NFT prompt you for it. When you enter the password at the prompt, it is not echoed on the screen. Because your password is not displayed on the screen, you have more control over who gets to see it. The following is an example eliminating the password:

```
BLD3"SMITH":: Return  
Password for BLD3/SMITH?
```


4.5.1 Defining Access-Control Information with NFT

There are two ways to specify access-control information during an NFT operation:

- By specifying access-control information on the command line
- By using the SET command

NFT temporarily stores access-control information for up to ten nodes.

In the following example, you access a remote node by requesting a directory listing of files located on the VMS node BLD3, user name SMITH, password OPEN.

```
NFT>DIRECTORY BLD3"SMITH OPEN": Return
```

NFT displays the list of files located in the requested directory including the block size of each file, and the time and date the file was last modified or created. Note that password information is shown in output as "password" to protect it from casual observation.

```
Directory of: BLD3"SMITH password":::SYS$SYSROOT:[SMITH]
FILE1.TXT;1      12      01-MAR-85  16:01:51
FILE2.TXT;1      34      02-SEP-83  14:20:35
FILE3.TXT;1      22      09-JUL-77  12:15:22
NFT>
```

NFT stores the access-control information for node BLD3 in its temporary table.

You use the SET command to specify and then save a user name, password and account, as well as specific disk and/or directory information. In the following example, you specify the user name and password for node BLD12. In addition, you specify the name of the subdirectory ARENA.FILES.

```
NFT>SET BLD12"ARENA BULL": Return
```

NFT stores the access information for node BLD12 in its temporary table. When you specify access information during an NFT operation and you are using the multiple command method, NFT temporarily stores the information for use in any following command dealing with the same node.

Access information saved during an NFT operation is deleted when you exit from NFT. However, if you saved the access information using NCP, it is not deleted.

If you often access a certain node and account, you can specify default access-control information using the NCP command **DEFINE NODE**. For more information about NCP commands, refer to the *DECnet-DOS Network Management Guide*.

4.5.2 Using Default Access-Control Information

After you specify access-control information by using either method, you can use the node name alone in subsequent file operations. NFT uses the access-control information you last specified for the node name. This is the **default** access-control information.

For example, if you request a directory listing of node BLD3 by using only the node name, NFT displays the same list of files as the first time you accessed the node. When you type a node name alone, you must follow the name with a double colon. For example:

```
NFT>DIRECTORY BLD3:: Return
Directory of: BLD3"SMITH password": :SYS$SYSROOT: [SMITH]

FILE1.TXT;1          12      01-MAR-85  16:01:51
FILE2.TXT;1          34      02-SEP-83  14:20:35
FILE3.TXT;1          22      09-JUL-77  12:15:22
NFT>
```

When you type a node name alone, NFT follows this procedure to determine the access-control information:

1. First, NFT checks to determine if you specified the access-control information earlier in the same NFT session. If you did, that access control information is used.
2. If you did not specify it earlier in the same session, NFT checks to determine if you specified it using NCP. If you did, that access-control information is used.
3. If you did not specify access-control information either using NCP or earlier in the same NFT session, NFT assumes that the remote node does not require the information and attempts the current operation.

4.5.3 Changing the Default Access-Control Information

You can change the default access-control information by typing different access-control information for the same node name in an NFT operation. In this case, NFT replaces the original access control information in its temporary table with the new information.

For example, if you request a directory listing of files on remote node BLD3, but specify a different user name and password, NFT displays a list of files for the new user:

```
NFT> DIRECTORY BLD3"DOC PEN": : Return  
Directory of: BLD3"DOC password": :SYS$SYSROOT: [DOC]  
  
ABC.TXT;1          10      07-SEP-85  12:10:49  
DEF.TXT;1          29      12-JAN-86  15:20:32  
GHI.TXT;1          47      03-MAR-86  14:10:05  
NFT>
```

Now if you request a directory listing using the node name alone, NFT displays this new list of files associated with the most recent access-control information for the node BLD3.

```
NFT> DIRECTORY BLD3: : Return  
Directory of: BLD3"DOC password": :SYS$SYSROOT: [DOC]  
  
ABC.TXT;1          10      07-SEP-85  12:10:49  
DEF.TXT;1          29      12-JAN-86  15:20:32  
GHI.TXT;1          47      03-MAR-86  14:10:05  
NFT>
```

You can also change the access-control information by using the SET command. In fact, when you use the SET command to specify default access-control information, you can replace only the information in the NFT table by using another SET command. For example, to change the access-control information for node BLD12, type:

```
NFT> SET BLD12"SMITH DANCE": : Return
```

Now if you request a directory listing by using the node name alone, NFT displays the new list of files associated with the most recent access-control information for the node BLD12.

```
NFT> DIRECTORY/BRIEF BLD12: : Return  
Directory of: BLD12"SMITH password": :SYS$SYSROOT: [SMITH]  
  
LAW.DOC;1          RULE.DOC;2  
NFT>
```


4.5.4 Displaying Access-Control Information

The SHOW command displays access-control information that you specify during an NFT session. For example:

```
NFT> SHOW Return
```

```
BLD3"DOC password"::  
BLD12"SMITH password"::
```

4.6 File Specifications

A complete file name is called a **file specification**. A file specification provides the computer system with all the information it requires to identify a unique file. Each operating system in the network has its own set of rules for naming files. Refer to Appendix A for more information about file specifications for other operating systems. For example, when you name a local DOS file, you must follow the standard DOS operating system format:

- A **drive name**. (This is optional if you are using the default drive name.)
- A **path name**. (This is optional.)
- A **file name** of up to eight alphanumeric characters.
- A **file type** of up to three alphanumeric characters, separated from the file name with a period (this is optional).

The following example specifies a file named SURVEY.CRD. The file is located on drive A in the subdirectory SUE, which is located in the subdirectory USERS:

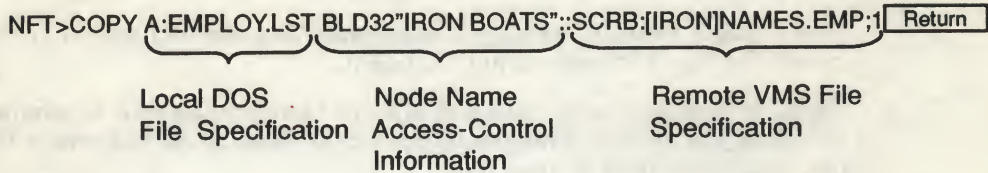
```
A:\USERS\SUE\SURVEY.CRD
```

When you access a file on a remote node, you must use a file specification that conforms to the conventions required by the remote node. For example, some operating systems accept a version number as part of the file specification:

```
BLD3::CHAP1.DOC;2
```

The example in Figure 4-2 shows how to copy a local DOS file called EMPLOY.LST to a remote VMS node named BLD32. The file is stored on the remote node as NAMES.EMP;1 (Version 1).

Figure 4-2: Copying a Local File to a Remote Node



4.7 Types of Files and Systems

Files can be one of two data types:

- **Image**

A file whose data is copied without any interpretation or data change. The file is copied and received as a copy of itself. The most common example of this type of file is an executable file. An image file is the same as a binary file.

- **ASCII**

A file whose records end with a carriage return/line feed (CR/LF) pair.

The algorithm that FAL uses to determine whether a file is image or ASCII (when copying the file from your personal computer to a remote node without switches) is the following: if the first 512 bytes contain a carriage return/line feed (CR/LF), the file is ASCII. If not CR/LF characters appear in the first 512 bytes, the file is image. When you use this rule, some image files might appear to be ASCII files.

Systems can be one of two types:

- **Stream**

With a stream system, a file is a series of continuous characters.

Most stream systems, including the DOS operating system, do not support such attributes as fixed or variable length records. If a file with these attributes is copied to the local system, the attributes are lost.

- **Nonstream**

With a nonstream system, file data is in specific record formats. Some examples of these formats are fixed length, variable length, and variable with fixed length control (VFC).

Fixed length records are all the same size. The size is fixed when you create the file, and you cannot change it.

Variable length records can be of different lengths, up to a maximum size that you specify. The maximum size is fixed when you create the file, and you cannot change it.

VFC records include a fixed length control field that precedes the variable length data. This format allows you to add data that labels the contents of the variable length portion of the record.

Refer to Table 4–1 for the changes that occur when you copy a file from a remote node.

Table 4–1: Copying Files from a Remote Node

Data Type	System Type	Record Attributes	How Stored
IMAGE	Not applicable	Ignored	As received from the remote node.
ASCII	Stream	Ignored	As received from the remote node with embedded carriage control.
ASCII	Nonstream	Other than implied CR/LF, PRN, or FTN.	As received from the remote node.
ASCII	Nonstream	Implied CR/LF pair	CR/LF added to each record.
ASCII	Nonstream	PRN or FTN	Data converted correctly.

NFT converts embedded carriage control characters by default. You can cancel this conversion with the /NOCONVERT switch.

Although NFT can usually determine whether a file is ASCII, you should use the /IMAGE switch when you copy a non-ASCII file to a remote node. Refer to Table 4–2 for the changes that occur when you copy a file to a remote node.

Table 4-2: Copying Files to a Remote Node

Data Type	System Type	Record Attributes	How Sent
ASCII	Stream	None	Unchanged. Records are determined by LFs.
ASCII	Nonstream	Variable, implied CR/LF	Carriage return/line feed pair is dropped. Records are determined by LFs.
IMAGE	Stream	None	Unchanged.
IMAGE	Nonstream	Fixed: 128 bytes	Unchanged.

4.8 File Operations

You can manipulate files on accessible DECnet nodes by using NFT.

You can include lists of up to ten file specifications in the APPEND, COPY, and TYPE commands. For example, you can append one to ten input files to a single output file. When you specify a list of files, the following rules apply:

- You cannot include more than 10 file specifications in a list.
- You cannot use wildcards (asterisk or question mark) in a list of files.
- You must separate items in the list with commas.
- When you enter a command line that contains more than one input file specification, these commands use temporary defaults. Temporary defaults are used for:
 - Node name
 - Device name
 - Directory name
 - File name and file type

If a file specification includes a node, device, or directory name, these names are applied to subsequent file specifications within the list. For example, the following command copies three test files from directory ADAMS on the device DBA1.


```
NFT> COPY RM2::DBA1:[ADAMS]TEST1.DAT TEST2.DAT,TEST3.DAT Return
```

4.8.1 Appending Files

To copy one or more files to the end of an existing file, use the APPEND command. You can append the following combinations of files:

- One or more local files to a remote file.
- One or more remote files to a local file.

NOTE

You cannot append a local file to a local file,
or a remote file to a remote file.

For example, to append the remote file B.TXT, located on the VMS node BLD3, to the local file A.TXT, type:

```
NFT> APPEND BLD3"SMITH OPEN":WRIT:[SMITH]B.TXT;3 A.TXT Return
```

The two files are now:

- The local file, A.TXT, which includes the contents of A.TXT and B.TXT.
- The remote file, B.TXT;3, which is the original copy of B.TXT on node BLD3.

If you are appending more than one file to the end of another file, separate the file specifications with a comma. For example:

```
NFT> APPEND BLD3"SMITH OPEN":WRIT:[SMITH]B.TXT;3,C.TXT;2,A.TXT Return
```

4.8.2 Copying Files

To copy files between your node and a remote node, use the COPY command.

When you copy a remote file to the local node, you must use the correct remote file specification. The following example copies a remote file named NEWS.DOC;10 from a VMS node called BLD3, to the local node. The remote file is located in the directory SMITH on device WRIT. The file is given the new name FLASH.DOC when it is copied to the local node.

```
NFT> COPY BLD3"SMITH OPEN":WRIT:[SMITH]NEWS.DOC;10 FLASH.DOC Return
```

You can copy a file from a remote node to a local node without specifying a local file name. For example:

```
NFT>COPY BLD3"SMITH OPEN": :WRIT:[SMITH]NEWS.DOC;10 Return
```

By default, the local file is named NEWS.DOC, the same as the original file.

When you copy a local file to a remote node, you must specify the local file name and the remote node name. You can also specify the remote file name in a format supported by the remote operating system. For example, to copy a DOS file called FILEA.TXT from drive B on the local node to directory SMITH on the BLD3 node, type:

```
NFT>COPY B:FILEA.TXT BLD3"SMITH OPEN": :WRIT:[SMITH]FILEA.TXT;3 Return
```

NFT truncates long file names and deletes special characters (to reduce file name size to eight characters and file type size to three characters) when copying files to your personal computer.

4.8.3 Deleting Files

To delete a file or a group of files, use the DELETE command. The following example deletes all versions of a file named TEST1.TST from a remote VMS node named RM104.

```
NFT>DELETE RM104::TEST1.TST;* Return
```

To delete only one file, enter the name and the type for the file. If there is only one version of that file on the remote node, NFT deletes it. You do not need to include the version number. If there are multiple versions of the same file, NFT deletes only the latest (or highest numbered) version of the file.

To delete a file that is not the latest version of that file, you must indicate the specific version number you want to delete. If the files FINDER.TXT;3 and FINDER.TXT;4 exist in your directory, you can delete FINDER.TXT;3, as follows:

```
NFT>DELETE RM102::FINDER.TXT;3 Return
```

This command deletes version 3 of the file FINDER.TXT on the remote node RM102.

4.8.4 Displaying Directory Information

To display a list of local or remote file specifications on your screen, use the **DIRECTORY** command. The names are displayed in the format used by the specified node. For example, to list file specifications from a directory on the remote VMS node BLD3:

```
NFT>DIRECTORY/BRIEF BLD3:: Return
```

```
Directory of: BLD3 "SMITH password":::SYS$SYSROOT:[SMITH]
```

```
APNDXA.DOC;3      CHAP1.DOC;2      CHAP2.DOC;13     DATA.DAT;9
NEWS.DOC;10       MEMO.TXT;1       TEST2.TST;6      TEST3.TST;2
TEXT.DOC;8
```

You can also either list a single file specification or use wildcards to specify a group of file specifications. The following example lists the group of files with a file type of TST:

```
NFT>DIRECTORY/BRIEF BLD3::*TST Return
```

```
Directory of: BLD3 "SMITH password":::SYS$SYSROOT:[SMITH]
```

```
TEST2.TST;6      TEST3.TST;2
```

4.8.5 Printing Files

To print a local file on a remote printer, use the **/PRINT** switch with the **COPY** or **APPEND** command. For example, to copy **FILEA.TXT** from the local node to the remote node BLD3 and then print the file at the remote node, type:

```
NFT>COPY/PRINT FILEA.TXT BLD3::FILEA.TXT;1 Return
```

Be aware that you cannot print a remote file on the local printer by using the **/PRINT** switch. If you try, the file is copied to the local node, and NFT displays the following warning message:

```
Warning: Cannot print files on local printer.
```

However, you can print a remote file on the local printer by copying it to the printer device.

To print a remote file on a remote printer, use the **PRINT** command. You must specify the file name that exists on the remote node. For more information about the **PRINT** commands, refer to Section 4.9.

4.8.6 Displaying a File's Contents

To display the contents of a file on your screen, use the **TYPE** command. For example:

```
NFT>TYPE RM6::PS:[SMITH]FILE1.DOC;2 Return
```

4.8.7 Running Command Files

A command file contains a list of command strings. To execute the commands within a remote command file on the remote node, use the **NFT SUBMIT** command followed by the name of the remote command file. By typing commands in one file, you can run the file many times without retyping the commands.

You can also use the **/SUBMIT** switch to execute commands in a local file on a remote node. This switch is valid only with the **APPEND** and **COPY** commands. When you use **/SUBMIT** with either of these commands, the copied file is queued to the remote node's batch system (after the copy operation is complete).

For example, the following commands are in a command file called **DUMP.CTL.13** on the VMS remote node **RM6**. The file is located on the device **PS:** in the directory **JONES**.

```
@PRINT PS:[SMITH]FILE1.DOC;2  
@PRINT PS:[SMITH]*.BAK
```

This file:

1. Prints the file called **FILE1.DOC;2** located in the **SMITH** directory.
2. Prints all files with a file type of **.BAK** in the same directory.

To run this command file, type:

```
NFT>SUBMIT RM6::PS:[JONES]DUMP.CTL.13 Return
```

The file is placed in the batch queue on node **RM6**.

4.8.8 Logging File Operations

When you enter an NFT copy command (such as COPY or PRINT) NFT displays a notification that it has opened the specified files. In the following example, NFT logs the copy operation and notifies you as it opens the specified files.

```
NFT>COPY INTRO.DOC BLD3"SMITH OPEN": Return
```

```
Copying file: INTRO.DOC to BLD3"SMITH password":DISK01:[SMITH]INTR  
O.DOC;1 [148 bytes at 9289 bytes/second]
```

```
NFT>
```

The notification line is displayed in two parts:

- The first part includes all the text up to the record count in brackets. It is displayed as soon as both the remote and local files are successfully opened.
- The second part is the record count (x bytes at x bytes per second) at the end of the line. This part is displayed when the copy operation is complete.

You can suppress the logging message for the APPEND, COPY, DELETE, PRINT, and SUBMIT commands by using the /NOLOG switch.

4.9 NFT Command Summary

NFT provides you with command verbs that allow you to manipulate files located on local and remote nodes. Table 4-3 lists each NFT command and its function. The remainder of the chapter discusses each command in alphabetical order.

Table 4-3: NFT Commands

Command	Function
APPEND	Appends files from the local node to an existing file on the remote node, or copies and appends files from the remote node to an existing file on the local node.
COPY	Copies files from the local node to the remote node or from the remote node to the local node.
DELETE	Deletes a local or remote file.
DIRECTORY	Lists files located in a specified local or remote directory.
EXIT	Exits from an NFT operation and returns control to the DOS operating system. (You can also exit by entering CTRL/Z Return .)
HELP	Displays information about NFT commands and switches.
PRINT	Allows you to queue a file that exists on a remote node to be printed at that remote node.
SET	Allows you to set remote file access defaults for the current NFT session.
SHOW	Displays access-control information for remote files.
SUBMIT	Submits a command file to be run on a remote node.
TYPE	Displays the contents of a local or remote file on the screen.

APPEND

APPEND

The APPEND command adds the contents of one or more input files to the end of an existing output file. You can append either ASCII or binary files to or from remote nodes. You can append the following combination of files:

- One or more local files to a remote file.
- One or more remote files to a local file.

You cannot append a local file to a local file, or a remote file to a remote file.

Depending on its position in a command, a file specification is either input (source) or output (destination).

Format

APPEND	[/ASCII: <i>[src-charset:dest-charset]</i> /DELETE /IMAGE /NOLOG /PRINT /STREAM /SUBMIT /UNDEFINED]	<i>input-file output-file</i>
---------------	---	--	---	-------------------------------

where

/ASCII: indicates that the file has records that end with a carriage return/line feed pair.
src-charset:dest-charset

You can also use the ASCII switch to perform character set conversion:

/ASCII:*src-charset:dest-charset*

where

src-charset and *dest-charset* are one of the following character set values:

DUTCH	FINNISH	FRCAN
FRENCH	GERMAN	ISO
ITALIAN	MCS	NORDAN
PORTUGUESE	SPANISH	SWEDISH
SWISS	UK	US

If you supply only one character set, it applies to the remote system. The local system uses the current character set.

/DELETE

requests that NFT delete the copied file from the remote node after the file is copied and printed. This switch is valid only with the /PRINT switch and only when appending files from a remote node.

/IMAGE

requests that NFT copy the file to the remote system as it is (binary data) with no conversion of any kind. The default record format is **FIXED**, and the default maximum record size (MRS) is 128 bytes. You can change these defaults with the **VARIABLE** (or **/VFC**) and **/MRS** switches. The last record can be shorter than the previous records.

/NOLOG

requests that NFT not print a notification line after the operation is complete.

/PRINT

allows you to print a file on the remote node's default printer after the file copy is complete.

/STREAM

allows you identify the type of stream file that you are transferring.

/SUBMIT

allows you to queue a command file to be executed at the remote node after the file copy operation is complete.

APPEND

<code>/UNDEFINED</code>	indicates that the file has an undefined record format.
<i>input-file</i>	specifies one or more input files to be copied. The file name consists of up to eight characters. The file type consists of up to three characters. If you specify multiple input files, you must insert a comma between the file specifications. NFT then appends the multiple files to the output file.
<i>output-file</i>	specifies the file to which the file(s) are to be appended.

In its longest form, a file specification consists of the node name, drive name, directory path, file name, file type, and version number. In its shortest form, a file specification consists of the a file name and file type. This form assumes that the file is in the current default DOS directory.

If you append more than one input file to create a single output file, the attributes of the output file are determined by the attributes of the first input file that you specify in the command. If the attributes of the input files differ, the append operation seems to succeed, but the output file might be incorrect.

Example

The following command appends the local file `FILE1.COM` to the remote file `FILE2.COM;1`, located at node `BLD14`. The resulting output file is printed.

```
NFT> APPEND/PRINT FILE1.COM BLD14::FILE2.COM;1 Return
```

You can also use the `/SUBMIT` switch with this command. `/SUBMIT` causes the file you just created to be queued to the remote node's batch system after the append operation is complete.

COPY

The COPY command creates a new file or a new version of a file at the destination node. You can use the COPY command to copy files from the local node to the remote node and from the remote node to the local node.

When using the COPY command, you should note the following:

- If a COPY command is issued to copy a file from a remote ULTRIX system to a personal computer (using a default output file specification) and the ULTRIX file specification contains directories, then the resulting personal computer file will be named incorrectly.
- If you copy a file from a remote system by using a wildcard character in the command, and if the wildcard matches a directory name, the directory file will be copied. However, it will contain no useful information.

Format for Copying Files to a Remote Node

COPY	/ALLOCATION= <i>number</i> /ASCII:[<i>src-charset:dest-charset</i>] /BLOCK /CC= <i>string</i> /DELETE /FIXED= <i>number</i> /IMAGE /LSA /MACY11 /MRS= <i>number</i> /NOLOG /NOSPAN /PRINT /SUBMIT /STREAM /UNDEFINED /VARIABLE /VFC= <i>number</i>	<i>input-file output-file</i>
------	---	-------------------------------

COPY

Format for Copying Files from a Remote Node

COPY $\left[\begin{array}{l} \text{/ASCII:[src-charset:dest-charset]} \\ \text{/BLOCK} \\ \text{/IMAGE} \\ \text{/NOCONVERT} \\ \text{/NOLOG} \end{array} \right] \text{input-file output-file}$

where

/ALLOCATION=number requests that the remote system set the allocation quantity (in blocks) for the file to *number* when creating a new file on a remote system. The default is 0, which causes allocation as needed. The valid range is 0 to 2147483647. This switch is valid when copying files to a remote system.

/ASCII:
src-charset:dest-charset indicates that the file has records that end with a carriage return/line feed pair.

You can also use the /ASCII switch to perform character set conversion:

/ASCII:src-charset:dest-charset

where

src-charset and *dest-charset* are one of the following character set values:

DUTCH	FINNISH	FRCAN
FRENCH	GERMAN	ISO
ITALIAN	MCS	NORDAN
PORTUGUESE	SPANISH	SWEDISH
SWISS	UK	US

If you supply only one character set, it applies to the remote system. The local system uses the current character set.

/BLOCK

copies files as image mode files, regardless of record structure. This allows you to move files that have undefined formats or files with very long record lengths. When you indicate **BLOCK**, it forces the following attributes: **IMAGE** mode, **FIXED** length, and an **MRS** of 512.

/CC=string

sets record attributes for files you are copying to RMS file systems only. If you are copying files to a stream system, such as TOPS-20, the record attributes are ignored. The variable *string* represents one of the following:

None	No record attributes.
FTN	Records contain FORTRAN carriage control.
CR	(Default) Records have an implied carriage return/line feed.
PRN	Records contain a fixed header with print carriage control. This is used with the /VFC switch.

It is assumed that if you specify an explicit record attribute for the output file, the file data already conforms to that specification.

/DELETE

is valid only with the **PRINT** switch. It requests that **NFT** delete the copied file from the remote node after the file is copied and printed. This switch is used only when copying a file to a remote node.

/FIXED=number

indicates that the records within the file are all the same length. By default, the length is 128 bytes, but you can change the length with the **/MRS** switch. This switch is only used when copying the file to a remote node.

COPY

/IMAGE	requests that NFT copy the file to the remote system as it is (binary data) with no conversion of any kind. The default record format is FIXED , and the default maximum record size (MRS) is 128 bytes. You can change these defaults with the /VARIABLE (or /VFC) and /MRS switches. The last record can be shorter than the previous records.
/LSA	indicates to a remote node that records are line-sequenced ASCII. This switch is valid only when copying files to a remote node.
/MACY11	indicates that the file is to be written on the remote node in MACY11 format. MACY11 files are valid only for remote systems.
/MRS=<i>number</i>	sets the maximum record size, where <i>number</i> is 0 to 32767. The default MRS for variable records is 0. This means there is no maximum for variable record file copies. The default MRS for fixed record files is 128. This switch is valid only when copying files to a remote node.
/NOCONVERT	requests that NFT not convert FTN or PRN carriage control characters in remote input files from RMS systems. If this switch is not used, the carriage control characters are converted. Use this switch with the COPY and APPEND commands only.
/NOLOG	requests that NFT not print a notification line after the operation is complete.
/NOSPAN	requests that records not span blocks. This switch is valid only for copying to RMS file systems.
/PRINT	allows you to print a file on the remote node's default printer after the file copy is complete. This switch is only valid when copying files to remote nodes.
/STREAM	allows you to identify the type of stream file that you are transferring.
/SUBMIT	allows you to queue a command file to be executed at the remote node after the file copy operation is complete.

/UNDEFINED	indicates that the file has an undefined record format.
/VARIABLE	resets the record format (RFM) to variable length records with a maximum record size of 0 when copying files to RMS file systems. You can change the maximum record size with the /MRS switch. This switch is only valid when copying files to a remote system.
/NFC=number	indicates that the file contains variable length records with fixed control headers. Where <i>number</i> (0 to 255 bytes) is the size of the control header when copying a file to an RMS file system. The default header size is 2. This switch is only valid when copying files to a remote node.
input-file	specifies the input file(s) to be copied. If you specify more than one input file, you must first specify the node, disk, drive, and pathname, and then the filenames separated by commas.
output-file	specifies the name(s) of the output file(s).

A file specification has one of two forms. In its longest form, it consists of a drive name, directory path, file name, and file type. In its shortest form, it consists of a file name and file type. This form assumes that the file is in the current default DOS directory.

You can use the COPY command to:

- Copy a single file – This operation copies a single file from one node to another. For example:

```
NFT>COPY SAM.TXT BLD3::SAM.TXT Return
```

- Copy a list of files – This operation copies more than one input file to the same number of output files.

For example:

```
NFT>COPY BLD3::SAM.TXT,SHAM.TXT SAM.TXT,SHAM.TXT Return
```

- Copy multiple files by using wildcards – This operation also copies more than one input file to the same number of output files. However, using this method, you can specify more than one file without typing more than one file specification. For example, the following command line copies all files with a file type of SUM from the local node to node BLD3. The files are copied with the same file names and file types.

COPY

```
NFT> COPY *.SUM BLD3: : *.SUM
```

If you are copying a local file to a remote node, you can include the /PRINT switch or the /SUBMIT switch. These switches cause the file you just copied to be queued to the remote node's printer or batch system, after the copy operation is complete.

If you specify multiple names, you must separate the file names by commas. There is a limit of 10 file names for each list.

Remember that you cannot use wildcards when you use file name lists.

Use caution when copying files from operating systems allowing long file names and nonalphanumeric characters. NFT truncates file names to eight characters and file types to three characters. For example, NFT copies a remote file named THISISMYFILENAME.RIGH THERE;1 as THISISMY.RIG.

Furthermore, the DOS operating system overwrites existing files of the same name. Therefore, if you copy a file named THISISMYOTHERFILENAME.RIGH THERE;1, NFT truncates it to THISISMY.RIG. DOS then overwrites the first file with the same name. When you copy multiple versions of the same file, DOS overwrites also.

Examples

The following command copies the local file PAGE1.TXT to the remote node BLD3. The file is printed on the remote printer and then deleted from the remote node.

```
NFT> COPY/PRINT/DELETE PAGE1.TXT BLD3: :PAGE1.TXT;1 Return
```

The output file specification can be omitted. In this case, the file specification is made the same as the input file specification. For example:

```
NFT> COPY/PRINT/DELETE PAGE1.TXT BLD3: : Return
```

DELETE

The **DELETE** command deletes one or more specified local or remote files. Be aware that some remote systems require a different file specification syntax from DOS. See Appendix A for details of foreign file specifications.

Format

DELETE [/NOLOG] *file-spec*

where

/NOLOG requests that NFT not print a notification line after the **DELETE** operation.

file-spec is any valid local or remote file specification.

Examples

The following command deletes the file **TAX.LST;3** from the remote VMS node **BLD3**.

```
NFT> DELETE BLD3::TAX.LST;3 Return
```

The following command deletes all versions of all files with a file type of **DOC** located on node **BLD6**.

```
NFT> DELETE BLD6::*.*.DOC;* Return
```

You can delete multiple files only by using wildcards. You cannot use a list of file specifications in the **DELETE** command.

DIRECTORY

DIRECTORY

The **DIRECTORY** command displays a list of local or remote file names, including the size (in blocks) and the time and date the file was last modified or created.

When using the **DIRECTORY** command, you should note the following:

1. If you use the **DIRECTORY** command to specify multiple subdirectories and some of those directories are protected, you will see an error message. However, the error message does not display the volume or the directory name.
2. When you use the **DIRECTORY** command with a remote node designation and no other information (such as **DIRECTORY REMNODE::**), NFT uses the default file specification of *.*. This produces the correct results on all remote systems except for **ULTRIX**. On an **ULTRIX** system, this command will list only the files that have a period in their names. To avoid this, issue the command using the following format:

```
DIRECTORY REMNODE::*
```

Format

DIRECTORY [**/BRIEF**] [**/FULL**] [*file-spec*]

where

/BRIEF causes NFT to display file names only.

/FULL causes NFT to display file names and file attributes.

file-spec is any valid local or remote file specification. If you omit the file specification, NFT assumes the local disk and the current default directory.

Examples

The following command displays the file named TAX.LST;4 located on node BLD3. The display includes the file's size and the time and date the file was last modified or created.

```
NFT> DIRECTORY BLD3::TAX.LST;4 Return
Directory of: BLD3 "SMITH password":::SYS$SYSROOT:[SMITH]
TAX.LST;4          20      09-JUL-89 12:30:52
```

The following command displays complete information for the file REPORT.DAT;4 on the remote node BLD3 including the following:

- The file name
- The size of the file
- The owner
- The type of file organization
- The type of record format used in the file
- The type of record attributes contained in the file
- The type of protection assigned to the file

All this information is displayed when you use the /FULL switch for files on a remote node. If you request a DIR/FULL listing for a file on the local node, you see only the size of the file and the time and date the file was last modified or created.

```
NFT> DIRECTORY/FULL BLD3::REPORT.DAT;4 Return
Directory of: BLD3 "SMITH password":::SYS$SYSROOT:[SMITH]
REPORT.DAT;4
Size:      8/9              Owner:   [910,20]
Created:    22-MAR-89 15:38:36
File organization:      Sequential
Record Format:          Variable length, maximum 128 bytes
Record Attributes:      Carriage return carriage control
File protection:        System: RWED, Owner:RWED, Group:RW, World:R
```

The following command lists all file names on the node BLD3. The list does not include sizes, times, or dates. If a file name consists of more than

DIRECTORY

19 characters, NFT truncates the name to 19 characters in the directory listing.

NFT DIRECTORY/BRIEF BLD3:: Return

Directory of: BLD3"SMITH password":SYS\$SYSROOT:[SMITH]

APNDXA.DOC;3	CHAP1.DOC;2	CHAP2.DOC;13	DATA.DAT;9
NEWS.DOC;10	MEMO.TXT;1	TEST2.TST;6	TEST3.TST;2
TEXT.DOC;8			

EXIT

The **EXIT** command causes you to exit from NFT and returns control to the DOS operating system.

Format

EXIT

Example

The following command causes you to exit from the NFT utility.

NFT> EXIT Return

You can also enter CTRL/Z and then Return to exit from NFT.

HELP

HELP

The **HELP** command displays information on your screen about NFT commands and switches.

Format

HELP { *command-verb* }
 /switch

where

command-verb is any valid NFT command verb.

/switch is any switch acceptable to NFT command verbs. It must be separated from the command by a space.

Examples

The following command displays a summary of all NFT commands.

NFT> **HELP** Return

Help is available on the following

commands:

APPEND	COPY	DELETE	DIRECTORY
EXIT	HELP	PRINT	SET
SHOW	SUBMIT	TYPE	

switches:

/ALLOCATION	/ASCII	/BLOCK	/BRIEF
/CC	/DELETE	/FIXED	/FULL
/IMAGE	/LSA	/MACY11	/MRS
/NOCONVERT	/NOLOG	/NOSPAN	/PRINT
/STREAM	/SUBMIT	/UNDEFINED	/VARIABLE /VFC

General command format is:

command /switches file file

Please note: Switches MUST always immediately follow the command.

Wildcards and file lists are supported for most commands.

NFT>

The following command displays a summary of the SHOW command.

```
NFT> HELP SHOW Return
```

```
SHOW
```

The SHOW command displays defaults for remote file access.

Example:

```
NFT>SHOW node::
```

```
NFT>
```

The following command displays a summary of the effects of the /ASCII switch.

```
NFT> HELP /ASCII Return
```

```
/ASCII
```

The ASCII switch indicates that a file contains ASCII text.

If neither the ASCII nor the IMAGE switch is used, NFT attempts to determine the file type automatically.

This switch is valid only with the COPY and APPEND commands.

```
NFT>
```


PRINT

PRINT

The PRINT command queues a remote file to be printed on a remote printer. (**Note:** The remote node cannot be another DECnet-DOS node and must support printing.)

You must indicate the file to be printed. The file is printed on the system default printer.

Format

PRINT { **/DELETE**
 /NOLOG } *file-spec*

where

/DELETE requests that NFT delete the copied file from the remote node after the file is copied and printed.

/NOLOG requests that NFT not print a notification line after the operation is complete.

file-spec is any valid remote file specification.

Examples

The following command queues the file TAX.LST;3 (located on the remote VMS node BLD3) to be printed on the default system printer of that node.

```
NFT> PRINT BLD3::TAX.LST;3 Return
```

The following command queues the file CHAPTER.MEM on node BLD6 to be printed on BLD6's default system printer. You cannot use wildcards or lists with the PRINT command.

```
NFT> PRINT BLD6::CHAPTER.MEM Return
```

SET

The SET command allows you to set default access-control information as well as disk and directory information for up to ten nodes. After you set this information, you can type the node name only, followed by two colons. NFT checks its default access-control information table for a record matching the specified node name. If it finds a match, NFT uses this information for network access.

Format

SET *node-spec*::

where

node-spec

is a remote node name and access-control information in quotes, followed by two colons.

Examples

The following command sets default access-control information for node BLD3.

```
NFT> SET BLD3"SMITH OPEN": Return
```

The following command sets default access-control information for node BLD1. This information includes only the user name JONES and the password NEW.

```
NFT> SET BLD1"JONES NEW": Return
```

SHOW

SHOW

The SHOW command displays the temporary default table of remote access information for the specified node. The password is not displayed on the screen. Instead, the character string *password* is displayed in its place.

Format

SHOW [*node-name::*]

where

node-name:: is a valid remote node name. If no node name is specified, NFT displays the contents of the access-control information table for all nodes known to the NFT temporary default table.

If NFT does not find the specified node name in the table, it checks to see if you typed default remote access-control information by using NCP. If you did, NFT displays the access-control information. If NFT does not find the node name in its own table or with NCP, NFT displays:

No defaults match: node-name::

Examples

The following command displays the contents of the access-control information table for all nodes known to the NFT temporary default table; that is, nodes you have accessed during this NFT session. The defaults you set using NCP are not displayed.

```
NFT> SHOW Return
```

```
BLD3"tower password"::  
RM4"strtfid password"::
```

The following command displays the access-control information for node BLD3.

```
NFT> SHOW BLD3: Return
```

```
BLD3"TOWER PASSWORD"::
```

SUBMIT

The SUBMIT command requests that the specified command file run on the remote node. The specified remote node must support command file submission and execution.

Format

SUBMIT [/NOLOG] *remote-filespec*

where

/NOLOG

requests that NFT not print a notification line after the SUBMIT operation.

remote-filespec

has two formats. In its longest form, it consists of node and access-control information followed by a file specification required by the remote node. (See Section 4.5 for details on access-control information.) In its shortest form, it consists of a node name followed by a file specification appropriate to the remote node. For example, **NODE::FILE.TYP;3**.

Command files contain one or more commands that are recognized and run by the remote node's operating system. This means that the format of the commands within the file must conform to the standards of the remote system.

Example

The following command requests that the file **BACKUP.COM** be run on the remote node **BLD1**. You cannot use wildcards or a list of file specifications in the SUBMIT command.

```
NFT> SUBMIT BLD1::BACKUP.COM Return
```

TYPE

TYPE

The **TYPE** command displays the contents of a local or remote file on your screen. You should use this command for ASCII files only.

Format

TYPE *file-spec*

where

file-spec is any valid local or remote file specification.

Example

The following command displays the contents of the file **PREFACE.DOC** located on the remote node **BLD3**.

```
NFT> TYPE BLD3::PREFACE.DOC Return
```

Using NDU to Access Remote Printers and Disks

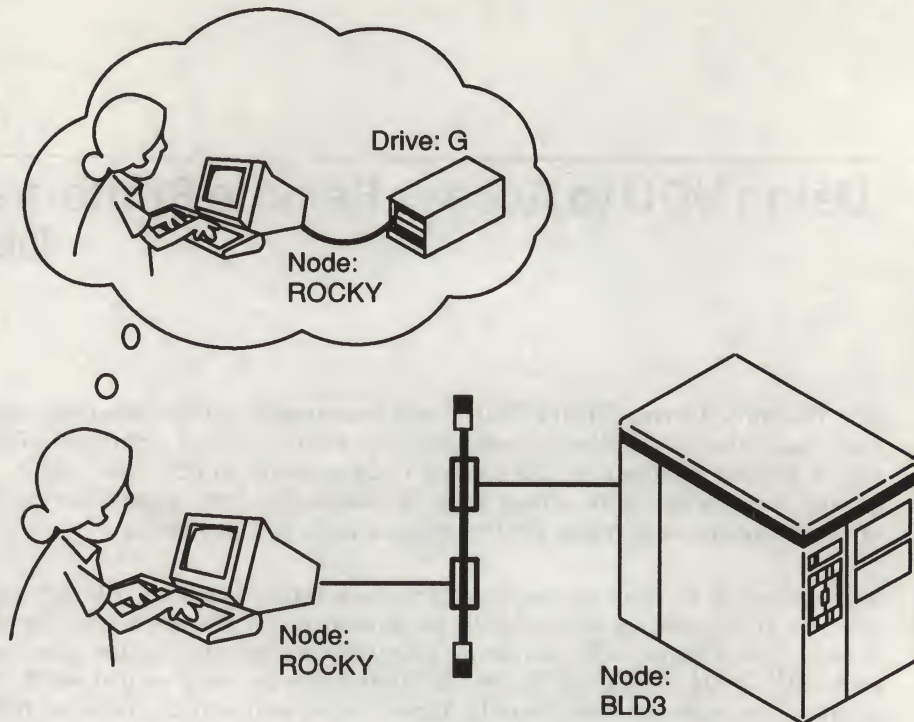
The Network Device Utility (NDU) lets you access remote printers and disks as if they were directly connected to your personal computer. The file or printer resident on the remote node appears to be a local device except for differences in access time. These differences depend on the type of communication services for the remote node you are using.

A remote disk file can be assigned a volume name (such as G:) and can be used by DOS utilities or applications software as if it were a local hard-disk volume (see Figure 5-1). A remote printer is assigned a device name (such as NPRN, PRN, LPT1, LPT2, or LPT3) and can be used as if it were a local printer. Because the devices only appear to be resident at the local node, they are called "virtual devices" (see Figure 5-2).

NDU provides virtual disk support at any remote DECnetTM node with file access server, such as the DECnet-DOSTM File Access Listener (FAL). Virtual disks are single remote files that appear to be entire DOS volumes. A single virtual disk can be used privately for read and write access, or it can be shared for read only access.

Any output directed to the specified printer device (such as NPRN) is forwarded to a remote system and retained until you issue an NDU command to close the file and queue it for printing.

Figure 5-1: Using Virtual Disks on DECnet-DOS

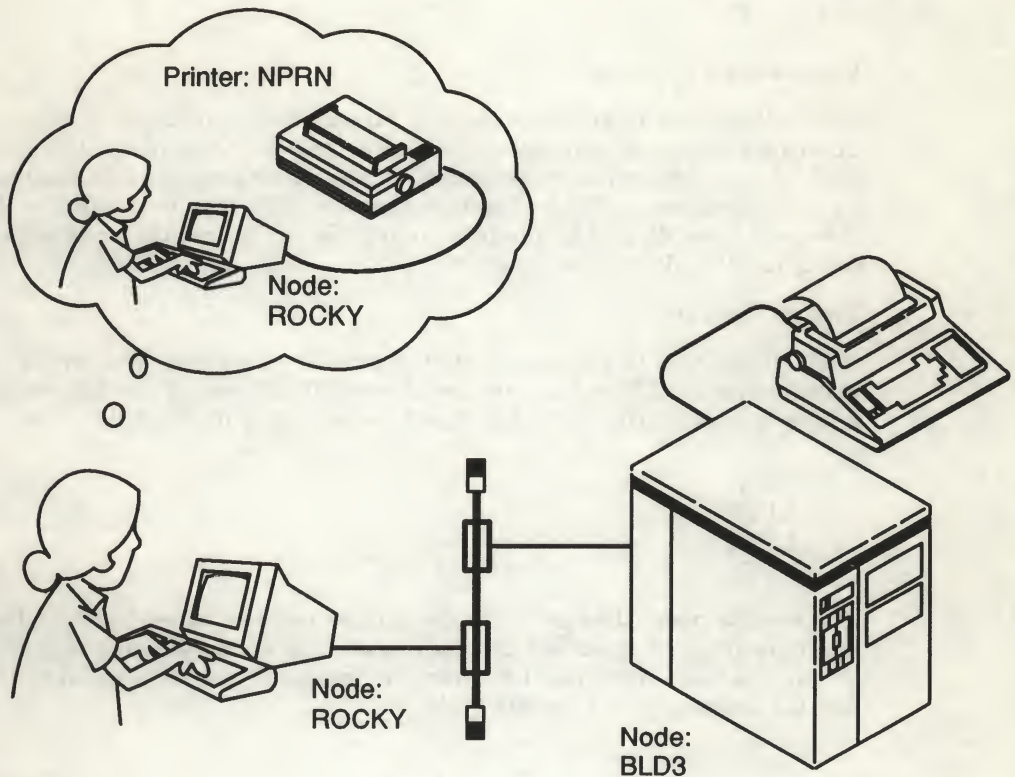


LKG-3203-891

Using NDU, you can:

- Assign a hard disk volume name to a new or existing remote file to be used as a virtual disk volume.

Figure 5-2: Using Remote Virtual Printers on DECnet-DOS



LKG-3204-891

- Stop the use of a remote file as a virtual disk volume.
- Delete a remote disk file that has been used as a virtual disk volume.
- Assign the printer device name NPRN to a disk file at a remote node.
- Stop the use of the virtual printer at the remote node and ask that the saved text file be queued for printing.

- Obtain status information about all assigned hard disk drives and the printer device.

NDU controls the use of both virtual disks and virtual printers in the following ways:

- **Virtual Disk Volumes**

NDU allows you to access up to four virtual disk volumes at a time. You can access four volumes on the same remote node, one volume on each of four different remote nodes, or any other combination totaling up to four volumes. Each remote volume is a file that is treated as if it were a hard disk. The file is a binary file on the remote node which contains DOS directories and files.

- **Virtual Printers**

NDU allows you to access one virtual printer at a time. The virtual print device NPRN can be changed by specifying one of the following device names on the command line in your CONFIG.SYS file:

PRN
LPT1
LPT2
LPT3

The remote node, that provides the virtual printer, saves any text that you output to the specified printer device in a temporary file on disk. When you close the virtual printer, the temporary file is queued to the default printer on the remote node.

The following topics are covered in this chapter:

- Installing the virtual device drivers (see Section 5.1).
- Running NDU (see Section 5.2).
- Entering NDU commands to control virtual disk volumes (see Section 5.3).
- Getting help (see Section 5.4).
- Special considerations while using NDU for virtual disks (see Section 5.5).
- An NDU command summary followed by a description of each NDU command and its syntax.

5.1 Installing the Virtual Device Drivers

Although NDU allows you to access virtual devices, two other programs called device drivers actually perform the input and output. The NDDRV.SYS device driver performs operations for virtual disks; the NPDRV.SYS device driver performs operations for virtual printers. You must install these device drivers on the DOS operating system to use NDU. If you do not, NDU displays an error message indicating that the disk driver or the printer driver is not installed. For example, the following message appears when the printer driver is not installed:

```
Network printer driver not installed
```

You install the device drivers when you first install the DECnet-DOS software. If you select this component as part of your configuration, the DECnet PCSA Client for DOS installation procedure automatically installs the device drivers. Refer to *Installing DECnet PCSA Client for DOS with Diskettes* for more details.

5.2 Running NDU

Using NDU you can create a disk file on a remote node for use as a virtual disk volume. You can create an unlimited number of virtual disks, but only four can be used at any one time.

NDU allows you to direct text to a remote node to be queued for printing. Enter the NDU utility by using one of the following methods:

- Enter a command string that includes NDU followed by the name of the specific NDU command, then press **Return**.

C:\>NDU command

After each NDU command is executed, the DOS prompt is returned.

C:\>

- Supply the name of the utility, NDU, and press .

C:\>NDU

NDU responds with a start-up message, then displays its own prompt. If you opened any virtual disk drives, NDU displays the status of the drives. For example:

Network Device Utility (NDU) 3.0
Network Disk Driver Version 3.0
Network Printer Driver is not installed

Disk Drive	Status/ Ndisk	Access User	Node/ User	Socket	Reads	Writes
G	CLOSED					
H	CLOSED					
I	CLOSED					
J	CLOSED					

Printer Device	Status/ Nprint	Node User	Socket/ User	Writes
LPT1	CLOSED			

NDU>

Enter the NDU command after the NDU prompt, and press . Continue this procedure until you have entered all the NDU commands you need.

NDU>command

NDU>command

NDU>command

NDU>

To exit from NDU, enter the EXIT command and press . The system returns with the DOS prompt.

NDU>EXIT

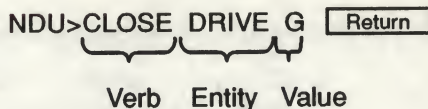
C:\>

You can also use to exit from NDU.

5.3 Entering NDU Commands

NDU commands consist of three parts; the command verb, the entity and a value. Figure 5-3 shows the command syntax.

Figure 5-3: Parts of an NDU Command



NDU commands act on entities and their values. When using an NDU command, you must provide the command verb, the entity, and the entity value.

5.3.1 Command Verbs

You can abbreviate command verbs to a character string that is unique to that verb. For example, you can type CRE for CREATE. For clarity and consistency, all examples in this chapter show the full command format.

NDU provides you with command verbs that allow you to access the remote files as virtual disk volumes. NDU also provides you with commands that allow you to direct text to a remote node to be queued for printing. Table 5-1 lists each NDU command and its function. The remainder of this section discusses each command in alphabetical order.

Table 5-1: NDU Commands for Virtual Disks

Command	Function
CLOSE	requests that an existing connection to a virtual hard disk be terminated. CLOSE also requests that NDU end an existing connection to a remote print file and queues the file for printing.
CREATE	instructs NDU to create and open a new remote file and format it to be treated as a virtual hard disk, or to create and open a remote printer.

Table 5-1 (Cont.): NDU Commands for Virtual Disks

Command	Function
DELETE	deletes a remote file that had been used as a virtual hard disk.
HELP	displays a message explaining the use of the NDU command and the meaning of the command parameters.
OPEN	requests a connection to an existing remote file.
SHOW	displays the activity and current state of all virtual hard disks and the virtual printer. A status report is displayed at the completion of other function requests.

5.3.2 Entities

Table 5-2 lists the entities that NDU commands act upon when controlling virtual disks or printers. When you specify more than one entity in a command line, use either a space or a tab to separate them.

Table 5-2: NDU Command Entities

Entity	Function
ACCESS <i>access-mode</i>	is used with the OPEN and CREATE commands to select the operating mode used to access the virtual hard disk. RW access indicates that the disk can be read from or written to, but it cannot be shared. RO access indicates that the disk can be shared but only for the purpose of reading.
ALLOCATION <i>n</i>	specifies the initial disk file size of the virtual disk file (in blocks) on the remote node. The value of <i>n</i> is a decimal number. This entity is valid only with the CREATE command.

Table 5-2 (Cont.): NDU Command Entities

Entity	Function
DRIVE <i>drive</i>	<p>specifies a drive name to be treated as a hard disk. The name is one alphabetic character. The drive name must be one of the names listed by the SHOW STATUS command.</p> <p>When you start NDU, it displays the drives that are available by indicating they are CLOSED. You can then select G, for example, as the drive name to be used in the DOS commands and functions that address a drive.</p>
MAX <i>xxx</i>	<p>is the maximum size of the virtual disk in megabytes. The value of <i>number</i> can be .360, .720, 1, 1.2, 1.44, 10, 20, or 32. The default is 32 megabytes.</p>
NDISK <i>file-name</i>	<p>identifies the file name associated with a drive name. For example, NDISK MYDISK1.DSK associates the file name MYDISK1.DSK with the drive name that is addressed by a DOS command. The file name can be any valid remote file name. It can include device and directory fields, provided it is properly formatted for the remote node's system type. If you include information for the device and directory fields, the total number of characters you can use in the command line is 127.</p>
NODE <i>node-name</i> or <i>node-address</i>	<p>identifies the virtual disk. The node name can be from 1 to 6 alphanumeric characters in length and must contain at least one alphabetic character.</p> <p>You can also specify a node address for the remote node. The node's address includes a unique area number and a node number, separated by a period:</p> <p><i>area.node</i></p> <p>where</p> <p><i>area</i> is a number in the range of 1 to 63.</p> <p><i>node</i> can be in the range of 1 to 1023.</p>

Table 5-2 (Cont.): NDU Command Entities

Entity	Function
NPRINT <i>file-name</i>	<p>specifies the name of the file which will contain the text sent to the virtual printer before it is queued for printing on the remote node. The file name can be any legal remote file name. It can include device and directory fields, provided it is properly formatted for the remote node's system type. If you include information for the device and directory fields, the total number of characters you can use in the command line is 127.</p>
USER <i>access-info</i>	<p>specifies access-control information for a specific remote node. This is the identification information that you enter during log-on at the hard disk's resident DECnet node. Each item in the identification string must be between 1 to 39 alphanumeric characters.</p> <p>The access-control string has the following format:</p> <p>USER <i>user-name</i> [/password [/account]]</p> <p>where</p> <p><i>user-name</i> is the name of the user on the specified remote node. If USER is not specified, default access-control information set by NCP will be used. You can set the default information by using the NCP DEFINE NODE command. For more detailed information about NCP commands, refer to the <i>DECnet-DOS Network Management Guide</i>.</p> <p><i>password</i> is the password you need to access files or programs on the specified node.</p> <p><i>account</i> is a 1 to 39 alphanumeric character string that consists of account information.</p> <p>Account information cannot be supplied unless you supply a specific password. If you supply a user name but you do not enter a password, you will be prompted for the password.</p>

Table 5-2 (Cont.): NDU Command Entitles

Entlty	Function
	To store the characters representing the access information in uppercase letters, type the characters. To store characters exactly as typed, place the string in double quotation marks. Character strings that are not within quotation marks are forced to uppercase.

5.4 Using the HELP Command

If you need assistance in selecting NDU commands, use the HELP command. Enter the HELP command, then press **Return**. For example:

NDU>HELP **Return**

The HELP command displays information about NDU commands and parameters

The format of the HELP command is:

HELP topic

Where topics include:

OPEN CLOSE DELETE CREATE SHOW EXIT HELP
 NODE USER DRIVE NDISK NPRINT MAX ACCESS ALLOCATION
 REDIRECTED

For information about one of the subjects listed, enter HELP plus a subject. Then press **Return**. For example:

NDU>HELP SHOW **Return**

5.5 Special Considerations While Using Virtual Disks

While using NDU to access virtual disks, you should be aware of the following information:

- When you issue an NDU CLOSE command for a remote virtual disk file residing on a personal computer that is running FAL, FAL reports an error message. This is only a warning; no error has occurred.
- The DELETE command fails if you create a virtual disk file with one NDISK file specification and then attempt to delete the virtual disk file with the following command:

DELETE NODE *node* NDISK *file*

You can delete the virtual disk, if it is connected, by using the file specification that you used to create it or using the drive letter.

- After you issue an NDU DELETE command, NDU prompts you for confirmation in the DOS style:

Are you sure (Y/N)?

CLOSE

The CLOSE command ends the connection between the specified drive and the virtual disk file on the remote node. The CLOSE command also ends the connection between the printer device and the file on the remote node. When you close a printer device, the remote file is queued to be printed on the remote printer. After the file is printed, it is deleted.

Format

```
CLOSE [ DRIVE drive
      NODE node-name NDISK file-name
      PRINTER ]
```

where

drive is one alphabetic character that specifies the virtual drive to close.

node-name identifies and locates the virtual hard disk's resident node. The node name must be from 1 to 6 alphanumeric characters, including at least 1 alphabetic character.

You can also specify a node-address for the remote node. Refer to Table 5-2 to see how the node address is formatted.

file-name specifies the name of the file that is the virtual disk on the remote node. It can include device and directory fields, provided it is properly formatted for the remote node's system type. If you include information for the device and directory fields, the total number of characters you can use in the command line is 127.

Examples

The following command ends the connection between the local node and the remote hard disk drive G.

```
NDU> CLOSE DRIVE G Return
```

Any subsequent operations to drive G result in the following error message:

CLOSE

Not ready error reading drive G

Abort, Retry, Ignore?

The following command ends the connection between the printer device and the printer on the remote node.

NDU> CLOSE PRINTER Return

The file on the remote node is printed and deleted. Any subsequent operations sent to the printer device result in the following error message:

Not ready error writing device NPRN

Abort, Retry or Ignore?

CREATE

The **CREATE** command creates and formats a new remote data file, which is treated as a hard disk. The **CREATE** command then establishes a connection between a volume name at your local node and the file.

For printers, the **CREATE** command creates a new virtual printer file and opens it for use. The printer file is created and then linked to the device **NPRN**.

Format

CREATE	{	NODE <i>node-name</i> NDISK <i>file-name</i>	[DRIVE <i>drive</i> USER <i>access-info</i> ACCESS { RW } ALLOCATION <i>number</i> MAX <i>xxx</i>]	}
		PRINTER NODE <i>node-name</i>	[NPRINT <i>file-name</i> USER <i>access-info</i>]	

where

NODE <i>node-name</i>	identifies and locates the virtual hard disk's resident node.
NDISK <i>file-name</i>	identifies the file name associated with a drive name.
DRIVE <i>drive</i>	specifies the name for a drive. The name consists of one alphabetic character. You should use the NDU SHOW command to list available names for the disk. Only the names for closed drives can be used. The default is the next available drive.
USER <i>access-info</i>	indicates access-control information for a specific remote node. If the access-control information is omitted, the access data specified with NCP is used instead. (See the entry under USER in Table 5-2 for a description of access-control information.)

CREATE

ACCESS
RW or RO

identifies the operating mode for accessing the virtual disk. RO specifies that the disk can be shared, but only for the purpose of reading. RW specifies that the disk can be read from or written to, but it cannot be shared. The default is RW.

ALLOCATION *number*

specifies the initial disk file size in blocks of the virtual disk file on the remote node. The value of *number* is a decimal number greater than the minimum virtual disk size. The default is the minimum size.

MAX *xxx*

specifies the maximum size to which the virtual disk can grow in megabytes. The value of *number* in this case can be .360, .720, 1, 1.2, 1.44, 10, 20, or 32. The default is 32.

The following message is displayed while NDU is creating a new virtual hard disk file. It is an informational message only. The number of blocks displayed on the screen changes as the file is being created.

```
CREATE DISK Writing block 0041
Disk   Status/Access   Node/   Socket   Reads   Writes
Drive  Ndisk   User
  G    OPEN    RW      BLD1    2   0   0
      [sam]temp.dsk   sam/password
```

Examples

The following command creates a new remote file named ACCTDEPT.TXT to be treated as a virtual hard disk, drive G.

```
NDU> CREATE DRIVE G NODE RM6 NDISK ACCTDEPT.TXT USER JONES Return
```

Future access to this hard disk might be restricted to USER JONES, depending on the specified access-control information. If DRIVE G already existed and is open, the following error message is displayed:

```
Drive "G" is already OPEN: CREATE function not done.
```

The following command creates a file named DEMO.TXT to be printed at the remote DECnet node RM6.

```
NDU> CREATE PRINTER NODE RM6 NPRINT DEMO.TXT USER JONES Return
```

DELETE

The DELETE command deletes an existing virtual disk file on a remote node.

Format

DELETE { **NODE** *node-name* **NDISK** *file-name* [**USER** *access-info*] }
 DRIVE *drive* }

where

NODE <i>node-name</i>	identifies and locates the virtual hard disk's resident node.
NDISK <i>file-name</i>	identifies the hard disk's file name.
USER <i>access-info</i>	indicates access-control information for a specific remote node. If the access-control information is omitted, the access data specified with NCP is used instead. (See the entry under USER in Table 5-2 for a description of access-control information.)
DRIVE <i>drive</i>	is one alphabetic character that specifies the virtual drive to delete.

After a DELETE request, NDU displays status information about all the drives. It also displays a message requesting verification that you want to delete the specified drive. For example:

Are you sure (y/n)?

The information is followed by the *NDU>* prompt.

Example

The following command deletes a file named TEMPDISK.TXT at node BLD9. If a drive was opened with the specified virtual disk, this command closes the drive. The drive is now available for use with another virtual disk.

NDU> DELETE NODE BLD9 NDISK TEMPDISK.TXT Return

HELP

HELP

The **HELP** command displays information about NDU commands and entities.

Format

HELP $\left[\begin{array}{l} \text{command-verb} \\ \text{entity} \end{array} \right]$

where

command-verb is the NDU command for which you want information.

entity is the NDU entity for which you want information.

Examples

The following command displays a list of all NDU commands and entities that you can get help on.

NDU> **HELP** Return

The **HELP** command displays information about NDU commands and parameters.

The format of the **HELP** command is:

HELP topic

Where topics include:

OPEN CLOSE DELETE CREATE SHOW EXIT HELP
NODE USER DRIVE NDISK NPRINT MAX ACCESS ALLOCATION
REDIRECTED

The system displays help information for the **CREATE** command.

NDU> **HELP CREATE** Return

THE **CREATE** command makes a new virtual disk and makes it ready for use. The format of the **CREATE** command for disks is:

CREATE DRIVE drive NODE name NDISK name USER user_string
ACCESS access ALLOCATION blocks MAX bytes

If the **DRIVE** parameter is omitted, the first free drive is used.

If the **USER** parameter is omitted, the string stored by NCP is used.

If the **ACCESS** parameter is omitted, **RW** (read/write) is used.

If the **ALLOCATION** parameter is omitted, the minimum size file is created.

If the **MAX** parameter is omitted, a value of 32 (32 million bytes) is used.

The CREATE command also makes a virtual printer ready for use.
The format of the CREATE command for the printer is:

CREATE PRINTER NODE name NPRINT name USER user_string

If the NPRINT parameter is omitted, a file name will be created.
If the USER parameter is omitted, the string stored by NCP is used.

OPEN

OPEN

The OPEN command establishes a connection between your local node and an existing virtual hard disk or printer at a remote DECnet node. If a printer is opened, the printer file is created and then linked to a printer device. Only one printer can be opened at a time. If a disk is opened, the existing file is linked to the device specified by DRIVE.

Format

$$\text{OPEN} \left\{ \begin{array}{l} \text{NODE } \textit{node-name} \text{ NDISK } \textit{file-name} \left[\begin{array}{l} \text{DRIVE } \textit{drive} \\ \text{USER } \textit{access-info} \\ \text{ACCESS } \left\{ \begin{array}{l} \text{RW} \\ \text{RO} \end{array} \right\} \end{array} \right] \\ \text{PRINTER NODE } \textit{node-name} \left[\begin{array}{l} \text{NPRINT } \textit{file-name} \\ \text{USER } \textit{access-info} \end{array} \right] \end{array} \right\}$$

where

NODE <i>node-name</i>	identifies and locates the virtual hard disk's resident node.
NDISK <i>file-name</i>	identifies the file name associated with a drive name.
NPRINT <i>file-name</i>	identifies the file name associated with the printer device.
DRIVE <i>drive</i>	specifies a drive name. The default is the next free drive.
USER <i>access-info</i>	indicates access-control information for a specific remote node. If the access-control information is omitted, the access data specified with NCP is used instead. (See the entry under USER in Table 5-2 for a description of access-control information.)
ACCESS RW or RO	identifies the operating mode for accessing the virtual disk. RO specifies that the disk can be shared, but only for the purpose of reading. RW specifies that the disk can be read from or written to, but it cannot be shared. The default is RW.

Following an OPEN request, NDU displays status information about the hard disk followed by the *NDU>* prompt.

Examples

The following command opens a connection between the local node and the remote DECnet node RM102. At that node, an existing file named ACCTDEPT.TXT is opened for READ WRITE access. All subsequent operations directed to the hard disk drive G are sent to ACCTDEPT.TXT at node RM102.

```
NDU> OPEN DRIVE G NODE RM102 NDISK ACCTDEPT.TXT Return
```

If the file does not exist, the OPEN operation fails, and the following error message is displayed:

OPEN function failed: the remote server could not find that file.

If the drive was already in use, the following error message is displayed:

Drive *name* is already OPEN, cannot OPEN it again.

If the file on the drive was already open for a create function, the following error message is displayed:

OPEN function failed: requested ACCESS conflicts with another's
use of the file.

The following command creates a file named LETTER.TXT to be printed at the remote DECnet node RM202.

```
NDU> OPEN PRINTER NODE RM202 NPRINT LETTER.TXT USER SAM Return
```

SHOW

SHOW

The SHOW command causes NDU to report on the status of all virtual disks and the virtual printer.

Format

SHOW STATUS

You can also display system status information through the use of DOS commands such as DIR and CHKDSK.

Example

The following command shows the status of all virtual disks and virtual printers.

NDU> SHOW STATUS Return

Sample output from this command appears as follows:

Disk Drive	Status/ Ndisk	Access	Node/ User	Socket	Reads	Writes
E	REDIRECTED					
F	OPEN	RW	RM102	6	241	68
	ND01.NVD		JONES/PASSWORD			
G	CLOSED					
H	CLOSED					

Printer Device	Status/ Nprint	Node User	Socket/ Writes
NPRN	CLOSED		

This display includes the following information:

- The name of the disk drive.
- The drive's status

OPEN—the drive is available for use.

CLOSED—the drive is closed and not available for use.

REDIRECTED—the drive can no longer be accessed by NDU because it has been reassigned for an alternate use.

- The file name.
- The access mode—read or write.
- The node name, the user name, and any account information.
- The number of the socket to which you are connected (DECnet assigns you to a socket).
- The number of completed read operations.
- The number of completed write operations.
- The name of the printer.
- The printer's status—OPEN or CLOSED.

The user and account information are displayed if you have specified them in the **CREATE** or the **OPEN** command. If you do not specify this information and it is obtained from the NCP database, it is not displayed.

Using DECnet-DOS Mail

The DECnet-DOS™ Mail utility lets you send messages and text files to users on other nodes or systems in the DECnet™ network. Note that Mail can only send messages and text to remote nodes. You cannot receive mail at your local node.

6.1 Running the Mail Utility

The first time you start the Mail utility, you are prompted for information. The Mail program uses this information each time it runs. The information is stored in the initialization file MAIL.DAT in the DECnet directory. If the file does not exist when you start Mail, the program asks you a series of questions and creates it with your answers. You can edit the information in MAIL.DAT at any time using a text editor. You can view the information in MAIL.DAT with the SHOW DEFAULTS command when you run Mail.

The initialization file can contain some or all of these mail parameters:

- `REPLY_ADDRESS = node-name::user-name`
- `DEFAULT_NODE = node-name`
- `CARBON_COPY = node-name::user-name`
- `EDITOR = path-name`
- `PERSONAL_MESSAGE = text-string`

The following sections describe these parameters in more detail.

6.1.1 REPLY_ADDRESS

A DECnet-DOS node cannot receive mail from other nodes. In order for users to respond to mail that is sent from your node, they must have a way to return or send messages to you. The DECnet-DOS Mail utility provides a way for other users to send mail to you on a different node by using the **REPLY_ADDRESS** parameter.

You must define the **REPLY_ADDRESS** parameter when you first run the Mail utility. If not, other users will not know where to send their replies. The address consists of a node name and a user name. The node name can be any node that you want to use for receiving mail. The user name should be a name that identifies you as a valid user on the specified node.

When other users receive mail from you, the **REPLY_ADDRESS** appears as part of the "SUBJ:" field in your message. The node name and user name are included in the "Reply to" field. Mail also includes the node address on the same line, in the "From" field just before the "Reply to" field.

For example:

```
FROM:  MSDOS::RM17::BROOKS
TO:    BLD16::PORTER
SUBJ:  GROUP MEETING "From 55.261, Reply to RM17::BROOKS"
```

In this example, the **REPLY_ADDRESS** is **RM17::BROOKS**. (The node number in this case is 55.261.) The same node name and user name appear in the first "FROM:" field. (MSDOS is the name of your local node.) The return address in the "FROM:" field can vary depending on whether or not your message had to be routed through another node. However, the "Reply to" field does not change. For this reason, the person responding to your message should always use the information in the "Reply to" field.

If you do not define a **REPLY_ADDRESS**, the "FROM:" field reflects only the default name **DECNETDOS**. For example:

```
FROM:  MSDOS::DECNETDOS
TO:    BLD16::PORTER
SUBJ:  GROUP MEETING
```

As indicated in this example, the user **PORTER** is not able to answer your mail message because the necessary information has been omitted. Be sure to include both a node name and your user name when you define the **REPLY_ADDRESS**.

6.1.2 DEFAULT_NODE

If you often send mail to the same node, you can shorten the address for that node by defining it as the `DEFAULT_NODE`. Once you have defined the `DEFAULT_NODE` in `MAIL.DAT`, you can send mail to any user on that node simply by specifying the user name. For example, you can define the `DEFAULT_NODE` as `BLD1` by entering it in `MAIL.DAT` as follows:

`DEFAULT_NODE = BLD1`

Whenever you send mail from your personal computer node to a user on `BLD1`, just include the user's name in the "TO:" field. For example:

```
MAIL>SEND   
TO:      SAM, JOHN, DAVID   
SUBJ:    Lab work
```

In this example, Mail first searches `MAIL.DAT` for the name of your `DEFAULT_NODE`. When it finds the name `BLD1`, it automatically sends your message to the users `SAM`, `JOHN`, and `DAVID` on node `BLD1`.

You can also use the `DEFAULT_NODE` for routing mail messages. If you want to send mail to a node that is not defined in your NCP database, the mail program uses the `DEFAULT_NODE` to route your message to the desired node. For example:

```
MAIL>SEND   
TO:      RM42::ANDERSON   
SUBJ:    Weekly report
```

In this example, the node `RM42` is a node that is not defined in the NCP database. In order to send your message to this address, the mail sender program requests that the `DEFAULT_NODE` (`BLD1`) forward the message to the user `ANDERSON` on node `RM42`.

If you do not define a `DEFAULT_NODE`, the mail sender program cannot route your mail messages. In addition, you cannot use the shortened form for mail addresses. You must include both the node name and the user name for any messages that you send.

6.1.3 CARBON_COPY

CARBON_COPY is the address to which a copy of your mail message is sent. Each time you send a mail message, a copy of that message is sent automatically to the CARBON_COPY address. This address requires both a node name and a user name. For example, you can define the CARBON_COPY address as RM204::BRIAN by entering it in MAIL.DAT as follows:

CARBON_COPY = RM204::BRIAN

Now, whenever you send mail to another user, a copy of that mail is automatically sent to the user BRIAN on node RM204. This node does not appear anywhere in your message. For example:

```
MAIL>SEND Return  
TO:      BLD7::PETER Return  
SUBJ:    Corrected problems
```

In this example, your message is sent to the user PETER on node BLD7 as well as to the user BRIAN on node RM204.

6.1.4 EDITOR

EDITOR specifies the complete path name for the editor that the mail program uses when you want to edit mail messages. You must include the /EDIT switch with a mail command if you want to edit a message or a text file before you send it.

6.1.5 PERSONAL_MESSAGE

The PERSONAL_MESSAGE is a message that other users see whenever you send them mail. The message can contain your name and phone number, or it can be any other descriptive text you wish to create. For example, you can create a message containing your name and phone extension by entering it in MAIL.DAT as follows:

PERSONAL MESSAGE = "Fred at ext 7227"

When you send a mail message, your personal message appears as part of the "FROM:" field. For example:

```
FROM:  MSDOS::RM17::BROOKS  "Fred at ext 7227"  
TO:    BLD16::PORTER  
SUBJ:  Group Meeting      "From 55.261, Reply to RM17::BROOKS"
```


6.2 Starting the Mail Utility

To start the Mail utility, enter a mail command. You can enter commands using either of two methods:

1. Enter a command directly from the command line. This method allows you to include one or more of the following qualifiers with your command:
 - The subject of your message (or a description of a file if you are sending one).
 - The name of a file (if you are sending one).
 - The address of the person who will receive your message or file.
2. Enter a command after the Mail prompt (MAIL>). This method allows you to include a subject, a file name, and an address, or to be prompted for this information. In addition, you have the option of editing the mail message or file before you send it.

You can also start the Mail utility from a window, if you are using MS-Windows. Mail runs under windows as a standard application. A standard application is an application that you can use with MS-Windows, even though it is not designed to do so. (There is a MAIL.PIF file included on your kit that allows you to run Mail as a standard application in windows.)

6.2.1 Entering Mail Commands from the Command Line

When you enter mail commands from the command line, you can include the name of a file that you want to send as well as the address of the person who will receive the file. For example:

```
C:\>MAIL ACCTRPT.TXT RM12::JONES Return
```

This command sends the file, ACCTRPT.TXT, to the user JONES on the node RM12.

You can also include a subject for the file you want to send. For example:

```
C:\>MAIL/SUBJECT="The accounting report" ACCTRPT.TXT RM12::JONES Return
```

If you do not include an address for sending a message, the program requests the information by issuing the "TO:" prompt. For example:

```
C:\>MAIL/SUBJECT="Here is your accounting report" ACCTRPT.TXT Return  
TO:
```

If you do not enter a subject, a file name, or an address, the Mail program responds with its prompt, MAIL>. For example:

```
C:\>MAIL 
```

```
MAIL>
```

6.2.2 Entering Mail Commands from the Mail Prompt

If the Mail prompt (MAIL>) is already on the screen, you can enter mail commands. You can type either MAIL or SEND to send a mail message. The program prompts you for an address and then a subject. For example:

```
MAIL>MAIL 
```

```
TO:      BLD6::BOOKER 
```

```
SUBJ:    Procedures 
```

You can also type MAIL or SEND followed by the name of a file that you want to send. The program prompts you for an address and a subject. For example:

```
MAIL>SEND RECEIPTS.DAT 
```

```
TO:      RM14::JONES 
```

```
SUBJ:    You need these figures for your report. 
```

6.3 Using Distribution Lists

Distribution lists transmit mail messages to a previously defined group of people. This saves having to retype a list of names each time you send mail to the people on the list.

If you frequently send mail to the same people, you can place their names in a distribution list file. When you then specify the list file as the destination for your mail message, the mail sender forwards the message to all of the names and addresses in the file. You can use this file as an addition to the names you have already indicated in the "TO:" field.

The distribution list file contains each user's mailing address. The address consists of the user's node name and user name. The node name and the user name are always separated by a double colon (::), and each address is on a separate line in the file.

The following is a sample distribution list:

```
BLD16::DEERE  
BLD4::JOHNS  
BLD8::M_OLDFIELD  
RM4::VOLKER  
RM6::G_HART
```

To create a distribution list file, use an editor that is installed on your system. (Refer to the appropriate user's guide for the type of editor you are using.) Enter the node names and user names of the people to whom you will be sending mail. Be sure to separate the node name and the user name with two colons (::), and enter each address on a separate line.

You can also supply a user name without the node name and double colon. If you do so, Mail will use the DEFAULT_NODE to send your message.

If you do not want mail to go to some of the people in your distribution list, you can edit the list to eliminate their names temporarily. To do this, simply place an exclamation mark (!) in front of their names. The next time you specify the distribution list, the mail program will skip over their names and will not send your message to those users. Later, if you want to change the list to include the people you previously excluded, simply edit the file and remove the exclamation marks.

The following is a sample distribution list file with two names temporarily omitted from the list.

```
!BLD16::DEERE  
BLD4::JOHNS  
BLD8::M_OLDFIELD  
!RM4::VOLKER  
RM6::G_HART
```

In this sample, the users DEERE and VOLKER on nodes BLD16 and RM4 will not receive your mail message.

To send mail using a distribution list file, enter the at sign (@) and the name of your distribution list file in the "TO:" field.

For example:

```
MAIL>SEND   
TO:      BLD9::MORGAN, BLD10::G_SMITH, @DEPART.LIS   
SUBJ:    Previous Month's Earnings
```

In this example, the name of the distribution list file is DEPART.LIS. While you can use several names and addresses in the "TO:" field, you can only include one distribution list file for each message you send. DECnet-DOS does not support multiple or nested distribution lists.

When using distribution list files, make sure the particular file you want to access is in the current directory. If it is not, then you must also indicate the name of the directory which contains the distribution list file you want to use.

6.4 Using Qualifiers

You can use the qualifiers /EDIT and /SUBJECT with the MAIL and SEND commands. These qualifiers can be used alone or combined. /EDIT lets you edit the mail message before you send it. This qualifier invokes the editor you have on your system. /SUBJECT lets you include the subject of your message at the same time you enter the mail command, which bypasses the need for the SUBJ: prompt. The following examples illustrate different possibilities for using /EDIT and /SUBJECT.

```
MAIL>MAIL/EDIT   
TO:      RM2::TAILOR   
SUBJ:    New information for you 
```

After you press (once you have completed the subject line), the program invokes the editor and clears the screen for you to enter the text of your message.

```
MAIL>SEND/SUBJECT="New information for you"   
TO:      RM2::TAILOR 
```

After you press (once you have completed the address line), the program invokes the editor and clears the screen so that you can enter the text of your message on the line immediately following the address.

```
MAIL>SEND/EDIT/SUBJECT="New information for you"   
TO:      RM2::TAILOR 
```

6.5 Getting Help for Mail

Help is available for the Mail utility by typing **HELP** at the mail prompt. For example:

```
MAIL>HELP Return
```

You can get help on the following topics:

EXIT	SEND
INIT FILE	SEND EXAMPLES
MAIL	SHOW DEFAULTS
MAIL EXAMPLES	

Information is also available for the following mail qualifiers:

/EDIT	/SUBJECT
-------	----------

To see **HELP** text for a specific topic, type **HELP** followed by the topic. For example:

```
MAIL>HELP INIT FILE Return
```

This command displays information about the initialization file.

6.6 Mail Command Summary

The Mail utility provides the following commands for sending mail messages:

EXIT	exits from the Mail utility.
HELP	displays on-line help for the Mail utility.
MAIL	allows you to send a Mail message to a remote non-DECnet-DOS node.
SEND	allows you to send a Mail message to a remote non-DECnet-DOS node.
SHOW DEFAULTS	displays the current values of the Mail environment.

The following sections describe each of the mail commands in alphabetical order.

EXIT

EXIT

The **EXIT** command causes you to leave the Mail utility.

Format

EXIT

Examples

In the following examples, the Mail program ends and you return to the prompt or drive you were using previously.

```
MAIL>EXIT Return
```

```
C:\>
```

or

```
MAIL> CTRL/Z Return
```

```
C:\>
```

HELP

HELP provides information about the Mail utility. **HELP** topics and qualifiers include:

/EXIT	SEND
INIT FILE	SEND EXAMPLES
EXIT	SHOW DEFAULTS
MAIL	/SUBJECT
MAIL EXAMPLES	

Format

HELP [*topic*]

Examples

To view the **HELP** file, type **HELP** at the mail prompt. For example:

```
MAIL> HELP Return
```

To view information for a specific topic, type **HELP** followed by the topic. For example:

```
MAIL> HELP SEND Return
```

This command displays information about how to send mail.

MAIL and SEND

MAIL and SEND

Both the MAIL command and the SEND command allow you to send mail messages and text files to other users in the network. These commands can be used interchangeably.

Note that you can send text files only using mail. You cannot send binary, 8-bit ASCII, or image files as mail messages. If you attempt to send these types of files, Mail will not display an error message. It will forward the files to the node you specify; however, the information in the files might be corrupted. Also, if you attempt to send a non-ASCII text file as a mail message, MAIL will not produce a useful error message. Instead, it will behave erratically.

Format

MAIL [/switch] [file-name]

where

/switch is a valid MAIL or SEND switch. The valid switches are:
/EDIT /SUBJECT

file-name is the name of a text file to be sent as a mail message.

To use the SEND or the MAIL command, type either command at the mail prompt. The Mail utility will first prompt you (with the "TO:" prompt) for the name of the user (or users) who will receive your message. You can include one or several user names, as well as the name of a distribution list file.

Examples

The following example requests a mail message to be sent to the user MORGAN on node BLD9, the user G_SMITH on node BLD10, and the addresses in the distribution list file DEPART.LIS.

```
MAIL> SEND Return
```

```
TO: BLD9::MORGAN, BLD10::G_SMITH, @DEPART.LIS Return  
SUBJ:
```

The utility prompts you for the subject of your message (with the "SUBJ:" prompt). You can avoid the "SUBJ:" prompt by specifying the /SUBJECT

qualifier when you first enter the command. For example:

```
MAIL> SEND/SUBJECT="Previous Month's Earnings" Return
```

```
TO: BLD9::MORGAN, BLD10::G_SMITH, @DEPART.LIS Return
```

The /SUBJECT qualifier can be followed either by a colon (:) or by an equal sign (=). The text of the subject must be enclosed in quotation marks (" ").

You can include a file specification with SEND or MAIL. This allows you to send a copy of a file to users on other nodes. The following command sends the file MARCH.DAT to the user GAMBOL on node RM3. The subject of the file is "Monthly Report".

```
MAIL> MAIL/SUBJECT="Monthly Report" MARCH.DAT Return
```

```
TO: RM3::GAMBOL Return
```

If you want to edit your message or file before you send it, use the /EDIT qualifier. /EDIT calls in the editor you specified when you first set up your Mail options. When you include this qualifier, the Mail utility clears the screen for you to enter the text of your message. If you are editing a file before sending it, the text of the file appears on the screen. When you have completed your edits, leave the editor. The mail message (or file) is then sent to all of the users you specified in the "TO:" field.

SHOW DEFAULTS

SHOW DEFAULTS

The SHOW DEFAULTS command displays the options that you have defined in your mail initialization file (MAIL.DAT). This command is useful for checking which of the mail parameters you have already defined, and whether they need to be changed or deleted.

Format

SHOW DEFAULTS

Example

The following command displays the information from MAIL.DAT.

```
MAIL> SHOW DEFAULTS Return  
  
REPLY ADDRESS = RM204::P JONES  
DEFAULT NODE = RM204  
CARBON COPY = RM204::P JONES  
EDITOR = EDT  
PERSONAL MESSAGE = PAJ on node BLD2  
MAIL>
```

File Specifications for Accessing Remote Files

A complete file name is called a file specification. A file specification provides your computer system with all the information it requires to identify a unique file. Each operating system in the network has its own set of rules for naming files.

A.1 Specifying Remote Files

DECnet-DOS™ can copy files to and from nodes running different operating systems.

Table A-1: Operating Systems and File Specifications

Operating System	File Specification
VMSTM	<i>dev:[dir]file-name.typ;ver</i>
TOPS-20TM	<i>dev:<dir>filename.typ.gen;att</i>
TOPS-10TM	<i>dev:filename.typ[p,pn,subdir]<prot></i>
RSXTM	<i>dev:[dir]filename.typ;ver</i>
DOS	<i>dev:dir\path\filename.typ</i>
ULTRIX-32TM	<i>dir/dir/.../file-name.typ</i>
ULTRIX-32mTM	<i>dir/dir/.../file-name.typ</i>
RSTS/ETM	<i>dev:[dir]file-name.typ</i>

NOTE

Most systems accept square brackets ([]) or angle brackets (< >) to delimit a directory name, and a period (.) or a semicolon (;) to delimit version. NFT accepts all these delimiters.

If any portion of these file specification formats is omitted, NFT assumes the default. File specifications of any format other than those listed in Table A-1 are considered foreign to DECnet-DOS NFT. When you type a foreign file specification, you must enclose it in quotation marks (" "). For example:

```
NFT>COPY RM2::"DK2:[100,100]NAMES.DAT" Return
```

This directs NFT to copy the RSTS/E file from the remote node RM2 to the local node, allowing unacceptable characters (such as a comma) in the directory name.

Most file specifications are not foreign.

A.2 Using Wildcards

Wildcards allow you to specify more than one file at a time. There are three wildcards you can use:

- A question mark (?) or a percent sign (%) matches any single character in the same position that the character occupies. For example, you can use TEST?.DOC to specify the following files:

TEST1.DOC
TEST2.DOC
TESTA.DOC
TESTB.DOC

- An asterisk (*) matches part or all of a file specification. For example, you can use *.DOC to specify the same list of TEST files. In the previous example, the asterisk could replace any file name with a file type of DOC.
- You can also use a combination of wildcards. For example, you can use TEST?.* to specify the following files:

TEST1.DOC
TEST2.DOC
TESTX.TXT

**TESTY.TXT
TESTZ.TRY**

Some remote systems do not support all three wildcards. Also, there might be wildcards which are supported on some remote systems but not supported by NFT.

Local and remote files can contain question marks, percent signs, or asterisks as wildcards. NFT cannot change the names of the files when wildcards are used. As an example, you can type:

```
COPY TEST?.* RM2::*.*
```

To rename the files included in the remote file specification, you cannot use wildcards. For example, you do not type:

```
COPY TEST?.* RM2::FARA?.*
```

Wildcards can be used with the **DELETE**, **TYPE**, and **DIRECTORY** commands. For example:

```
DELETE BLD3::*.*.*
```

This command deletes all files in the user's default directory on the node BLD3.

THE UNIVERSITY OF CHICAGO
DIVISION OF THE PHYSICAL SCIENCES
DEPARTMENT OF CHEMISTRY

REPORT OF THE
COMMISSIONERS OF THE
UNIVERSITY OF CHICAGO

FOR THE YEAR
ENDING JUNE 30, 1900

CHICAGO
PUBLISHED BY THE
UNIVERSITY OF CHICAGO PRESS

1901

NFT Error Messages

When an NFT error occurs, you should receive one or more of the error messages listed in this appendix.

If you are accessing a remote system and the remote node reports an error for which no text has been defined, a Data Access Protocol (DAP) message similar to the following is displayed:

DAP error reported by remote node: *error-code/error-code*

The error message consists of a pair of DAP error codes in octal. The first error code indicates a specific error type. The second error code identifies the specific reason for that error. To determine the meanings of the displayed error codes, refer to the appropriate appendix in the *DECnet-DOS™ Programmer's Reference Manual*.

If a local error occurs for which there is no defined text, a message similar to the following is displayed:

Unexpected network error, ERRNO value: *n*

where *n* is in decimal.

If you receive either of these messages, see your network coordinator.

B.1 Types of NFT Error Messages

NFT error messages fall into three general categories:

- Network errors
- Command line error messages
- File input/output (I/O) errors

Command line error messages

These messages are displayed when you mistype part of a command line (such as the verb or a switch).

For example:

```
NFT>COPI PAGE1.TXT BLD3::PAGE1.TXT;1 Return
```

Error: Unrecognized command: COPI

The COPY command is mistyped.

File input/output (I/O) errors

These messages are displayed when you are accessing a local or remote file or submitting a remote file. Remote file I/O error messages are displayed on two lines:

- The first line states the problem.
- The second line explains why the problem occurred, and the line begins with the word "Because." These messages are referred to as secondary messages and are listed at the end of this appendix.

For example:

```
Error Opening File: BLD3::MYSTAT.DAT  
Because cannot assign address - possibly node name is undefined.
```

You typed a remote node name that was not previously defined.

Network errors

These messages are displayed when you try to run NFT on a system where network support or some part of it is missing or not working properly.

Within each category, the messages are listed in this appendix alphabetically.

When a command line specifies multiple operations and an error is encountered during one operation, NFT tries to execute subsequent operations even after the error has been detected.

B.2 Command Line Error Messages

The error messages listed in this section are all related to the syntax of a command line.

Error: Cannot APPEND to list of files.

You tried to append one or more files to the end of more than one file. You can have only one output file.

Error: Cannot APPEND to wildcard file name.

You included a wildcard in the output file name when you tried to append an input file to an output file.

Error: Cannot COPY from wildcard specification to explicit files.

You tried to copy multiple files by using a wildcard while you wanted the output files to have specific names or file types.

Error: Cannot SUBMIT list of files.

You tried to submit more than one remote command file (in a list) to be run on the remote node.

Error: Command string too long.

The command line you typed was longer than 256 characters.

Error: File lists not supported for DELETE command.

You tried to delete a list of files. You can delete multiple files using wildcards.

Error: File lists not supported for DIRECTORY command.

You tried to display a directory of a list of files. You can display multiple file names using wildcards.

Error: Illegal ALLOCATION quantity: *text*

You used the /ALLOCATION switch with the COPY command and specified a nonnumeric value.

Error: Illegal Maximum Record Size: *value*

You used the /MRS switch with the COPY command and specified either a nonnumeric value or a value outside of the range of 0 to 1024.

Error: Illegal VFC amount: *value*

You used the /VFC switch and specified either a nonnumeric value or a value that is less than 0 or greater than 255.

Error in allocation amount.

You used the /ALLOCATION switch without a value. Please supply one.

Error in allocation quantity.

You used the /ALLOCATION switch with the COPY command, and the remote node did not accept this value.

Error in maximum record size.

You used the /MRS switch with the COPY command, and the remote node did not accept the value you specified.

Error in VFC size.

You used the /VFC switch with the COPY command and specified a fixed length header, which was not accepted by the remote node.

Error: Quoted local files are illegal.

You enclosed a local file specification in quotation marks (" "). Quotation marks are used only for remote file specifications.

Error: Too many file names in list. Limit is 10.

You included more than ten file names in a list (for example, with the APPEND command).

Error: Two paths in a row.

You specified more than one path name without specifying the corresponding file names in between.

Error: Unrecognized command: *command*

You mistyped an NFT command or tried to use a command that NFT could not recognize.

Error: Unrecognized /CC option: *option*

You used the /CC switch and specified an incorrect record attribute. The attributes must be one of the following:

- None - No attributes
- FTN - FORTRAN carriage control
- CR - Implied carriage return/line feed
- PRN - Fixed header with carriage control

Error: Unrecognized switch: *switch*

You mistyped the name of a switch or used a switch that NFT could not recognize.

Error: Wild cards are not supported in file name lists.

You tried to list files by using wildcards (for example, with the APPEND command).

B.3 Warnings

The following messages are warnings. When they are displayed, the current operations will continue.

Print switch not supported with this command.

You used the /PRINT switch with a command that does not support the switch (for example, DELETE).

Warning: Cannot DELETE both remote and local files.

xxx file name(s) will be ignored.

You specified a local and a remote file to be deleted. NFT deletes the file you listed first.

Warning: Cannot perform DIRECTORY for both local and remote files.

xxx file name(s) will be ignored.

You requested a directory listing of both remote and local file names. NFT displays the directory you listed first.

Warning: Cannot print file(s) on local printer.

You included the /PRINT switch in a COPY operation from a remote node to the local node. You cannot print remote files on the local printer. (However, you can print local files on a remote printer.)

Warning: Cannot TYPE from one file to another.

You included two file specifications with the TYPE command.

Warning: Renaming files with wildcard specifications not supported.

xxx file name(s) will be ignored.

You tried to rename a file with the COPY command and used a wildcard in both the input file name and the output file name.

Warning: File(s) will NOT be deleted after copy.

The /DELETE switch was used as an invalid switch with the COPY command. /DELETE is valid only with /PRINT.

Name is too long for local file. Limit is 12 characters.

Enter local file name:

You entered a file name that is too long for DOS to accept. The local file name must be 12 characters or less.

Unable to make local file name from remote name: *node-name::file-name.typ*
Enter local file name:

You omitted the local file name with the COPY command, and NFT was not able to create a local file name from the specified remote file name.

B.4 File I/O Error Messages

The messages listed in this section are all related to accessing local and remote files or submitting a remote command file. If you receive a remote node error message, refer to the documentation for that operating system, or see your network coordinator for instructions on how to correct the problem.

Examples of file I/O errors are improper file specifications or a file read error. Remember that remote file I/O messages are displayed on two lines. The second line is a secondary message that describes why the error occurred. Many of the secondary messages can be displayed with more than one file I/O message.

Cannot delete file: *file-name.typ*

NFT cannot delete the local file you specified because it is a directory file or has a Read Only protection.

Error Closing File: *node-name::file-name.typ*

NFT cannot close the remote file you specified. The secondary message explains the specific reason.

Error Deleting File: *node-name::file-name.typ*

NFT cannot delete the remote file you specified. The secondary message explains the specific reason.

Error In Directory of *node-name::*

There is a problem with the remote directory you specified. The secondary message explains the specific reason.

Error Opening File: *node-name::file-name.typ*

NFT cannot open the remote file you specified. The secondary message explains the specific reason.

Error Reading from: *node-name::file-name.typ*

NFT cannot copy from the remote file you specified. The secondary message explains the specific reason.

Error Submitting File: *node-name::file-name.typ*

NFT has a problem running the remote command file you specified. The secondary message explains the specific reason.

Error Printing File: *node-name::file-name.typ*

NFT has a problem queuing the remote file to a printer. The secondary message explains the specific reason.

Error Writing to: *node-name::file-name.typ*

NFT cannot copy to the remote file you specified. The secondary message explains the specific reason.

Error: Cannot SUBMIT local file.

You tried to run a local command file.

Error: Unable to type non-ASCII file: *file-name.typ*

You tried to type a file that has a non-ASCII format. NFT displays the name of the file you tried to type.

File not found: *file-name.typ*

You tried to access a file that NFT could not find. The file might not exist, or you mistyped the file name.

Record too big - Try /BLOCK or /IMAGE

The usual cause is attempting to transfer a binary file as ASCII. Use the /BLOCK switch to transfer stream files that were put on the remote system with PCSA, and use the /IMAGE switch to create fixed record length files on the remote system.

Unable to open file: *file-name.typ*

NFT cannot open the local file you specified.

B.5 Network Error Messages

The error messages listed in this section are related to network software, including temporary files created by NFT.

Network errors pertain to network operation failures or rejections. For example, connection requests can be rejected by the network because of insufficient network resources or to an invalid node name.

Allocation quantity too large.

The disk on the remote system is too full to process your request.

Connection lost.

The remote system broke the connection.

Error opening file {remote file spec} because local node is off.

You should check to make sure that the state of the local node, circuit, and line is ON.

Host is down.

You are trying to access a remote host, and it is not in operation.

Insufficient network resources.

The remote system could not accept any more connections.

Network is down.

You are trying to perform network activity, and the network is not currently available.

Network not installed.

You must start DECnet to use NFT.

No route to host.

This message is an indication of one of the following conditions:

- The network driver is not loaded, or you are not using the correct version.
- The line state is OFF.
- The remote host is currently unreachable.
- Problem with temporary file for remote file names.
- You are copying a file from a remote node, and you included a wildcard in the command. For example, COPY NODE::*.*. NFT first requests a directory list of the remote files. NFT then stores these file names in a temporary local file on the default disk. For example, if a problem occurs with the temporary file because of lack of disk space, this message is displayed.

Remote system DAP buffer size < 256.

Two programs on different systems have such different buffer sizes that they cannot communicate with one another.

B.6 Secondary Error Messages

Secondary error messages are displayed with remote file I/O messages. Many of the secondary messages can occur with more than one file I/O message. Also, a file I/O message can be displayed with one of several secondary messages depending on the cause of the problem.

Because bad record size.

The specified record size is either invalid or illegal for the specified operation.

Because cannot assign address, possibly node name is undefined.

The network rejected an attempted connection because the remote node name did not correspond to any node name defined at the local node.

Because cannot open file.

An error occurred on a file open operation.

Because cannot position to end of file.

NFT cannot append the specified input file to the end of the specified output file.

Because device is write locked.

The specified output file cannot be created because the output device is write locked.

Because directory full.

You tried to copy a file to a remote node, and the remote directory is already full.

Because directory not found.

The specified directory does not exist on the node/device specified or defaulted in the file specification.

Because disk quota exceeded.

You tried to perform a remote file operation, and there is no room on the remote disk.

Because error in directory name.

The specified directory does not conform to the syntax of the target system.

Because error in file name.

The specified file name does not conform to the syntax of the target system.

Because error in record attribute.

You specified remote file record attributes (such as FORTRAN, implied line feed/carriage return, embedded or VMS print file) that are not valid.

Because error in record format.

You specified a remote file record format (such as fixed, variable, VFC, or stream) that is not valid.

Because file locked by other user.

Another user currently has the specified file locked or open for writing. You can also receive this message in a full directory listing.

Because file not found.

The specified file or files do not exist.

Because file read error.

An irrecoverable error has occurred.

Because file write error.

An irrecoverable error has occurred.

Because Illegal record attributes.

The file's record attributes (that is, FORTRAN, implied CR/LF, embedded, or VMS print file) are invalid or unsupported by NFT.

Because Incorrect user access Information.

The network rejected an attempted connection because the specified access control information (user ID, password, and account) does not match a valid account on the remote node.

Because Invalid record format.

The file's record format (that is, fixed, variable, VFC, or stream) is invalid or unsupported by NFT or the remote FAL.

Because Invalid wildcard operation.

The remote system rejects the specified wildcard as inappropriate for the specified operation.

Because privilege violation.

An operation was specified for which you do not have privileges.

B.7 Other Error Messages

Address already in use.

Address family not supported by protocol family.

Argument list too long.

Argument too long.

Attribute read error.

Attribute write error.

Bad address.

Bad block on device.

Bad file number.

Bad terminator or bad value for switch:

Bad version number.

Cannot close file.

Cannot get JFN for file.

Cannot open local file with DOS reserved names:

Cannot PRINT list of files.

Cannot PRINT local file.

Connection refused.

Connection reset by peer.

Connection timed out.

Destination address required.

Device not available.

Device not found.

Device not ready.

Device or file full.

Disk full or other error in closing file.

Disk full or other error in writing file.

Disk usage exceeds quota.

Error in data type.

Error in file type extension.

Failure to get transmit buffer.

F11-ACP could not access file.

F11-ACP could not create file.

F11-ACP could not mark file for deletion.

File activity precludes operation.

File already open.

File exists.

File extend failure.

File is currently in an undefined state.

File name syntax error.

File name too long.

File not found.

FSZ field invalid.

Illegal record encountered.

Invalid DAP message type received.

Invalid DAP message format received.

Invalid file options.

Invalid object name format.

Invalid wildcard context value.

Logical name error.

LPT page limit exceeded.

Message too long.

Network capacity exceeded.

Network dropped connection.

Network operation failed at remote node.

Network operation not supported.

Network operation timed out.

NFAR error, FFIRST with dir in progress.

NFAR error, file not open.

NFAR error, FNEXT with no dir in progress.

NFAR error, illegal access.

NFAR error, invalid function.

NFAR error, invalid RATs.

NFAR error, missing ::.

NFAR error, missing quote.

NFAR error, no \\\\ or *\\.

NFAR error, no more room.

NFAR error, record buffer too small.

NFAR error, socket not found.

No buffer space available.

Node name cannot be longer than 6 characters.

Node is unreachable.

No more sockets available.

No such device.

Not supported by remote.

Operation illegal or invalid for file organization.

Operation in progress.

Operation not supported on socket.

Operation would block.

Parity error on device.

Path and file names not valid for SET, SHOW, or EXIT commands.

Protocol not available.

Protocol not supported.

Remote object is too busy.

Result too large.

Socket is already connected.

Socket is not connected.

Socket operation on nonsocket.

Socket type not supported.

Spool or submit command file error.

Switches not valid for SET, SHOW, or EXIT commands.

Task not available.

Too many open files.

Unexpected DAP message received.

Unmatched quotes:

Unrecognized local file specification, remote-to-remote operations not supported.

Unrecognized object.

Unrecognized remote file specification:

Unrecognized remote file specification, local-to-local operations not supported.

Unsupported DAP flag field received.

VFC value must be less than 256.

B.7.1 Warnings

Extra local file names will be ignored.

Extra remote file names will be ignored.

Warning: Destination file name list longer than source list.

NDU Error Messages

This appendix lists the error messages you might encounter when using the NDU utility. The first section deals with Virtual Disk error messages; the second section lists Virtual Printer error messages. The NDU error messages fall into three categories and are listed alphabetically within each category.

Command line error messages

These messages are displayed when you mistype part of a command line. The error message includes a circumflex (^) which appears directly under the error, followed by explanatory text. For example:

```
NDU>CLOSE DRIVE ONE Return
```

^ Disk DRIVE must be a single character.

Network error messages

These messages are displayed when you try to run NDU on a system where network support or some part of it is missing or not working properly.

Execution error messages

You might encounter these messages when you attempt to perform an NDU function, such as CREATE or OPEN.

C.1 Virtual Disk Error Messages

The following error messages might occur when you perform operations on virtual disks.

C.1.1 Command Line Error Messages

The error messages listed in this section are related to the syntax of a command line.

ACCESS must be either RO or RW.

You can specify only RO or RW as valid ACCESS options.

DRIVE *name* is too long. It must be a single character.

A disk drive's name cannot exceed one alphabetic character.

Initial ALLOCATION must be a decimal number between 0 and 65535.

You can specify a decimal value in the range 0 to 65535. However, because of internal requirements, NDU always sets this value greater than 30 when the file is actually created.

Unexpected syntax error.

You mistyped a command.

Unrecognized command.

You mistyped an NDU command or tried to use a command that NDU could not recognize.

Unrecognized HELP option; try typing just HELP.

You typed an invalid HELP option. Just type HELP for assistance.

Unrecognized NDISK file name.

A valid file name consists of 1 to 126 alphanumeric characters. The file name can include device and directory fields, but it must be in the proper format for the remote node's system type.

Unrecognized node name.

A valid node name consists of 1 to 6 alphanumeric characters. It must contain at least one alphabetic character. The node name can also be a unique numeric address in the format of *area.number*.

Unrecognized NPRINT file name.

A valid file name consists of 1 to 126 alphanumeric characters. The file name can include device and directory fields, but it must be in the proper format for the remote node's system type.

Unrecognized Parameter.

You supplied an invalid parameter with an NDU command. Type **HELP** for assistance.

Unrecognized SHOW option.

NDU could not recognize the option. The only valid option for the **SHOW** command is **STATUS**.

Unrecognized USER name.

NDU could not recognize the **USER** name. A valid **USER** name consists of access-control information: *user-id*, *password*, and *account*. Each field can include 1 to 39 alphanumeric characters. Enter a valid **USER** name.

C.1.2 Network Errors

A network error consists of a pair of messages separated by a colon. The message format is:

ndu-specific-message : *dap-specific-message*

The first message can be one of the following NDU messages:

CLOSE during DELETE function failed.

CLOSE function failed.

CREATE function failed.

DELETE function failed.

OPEN function failed.

The second half of the message string can be one of the following DAP messages:

A received DAP message was poorly formed.

During the dialog with the File Access Server on the remote node (which was necessary to execute your request), a message was received of the proper type, but it did not contain the proper contents.

Action: For an explanation of the DAP message, refer to the *DECnet-DOS Programmer's Reference Manual*. For additional assistance, see the person responsible for your network.

An unexpected DAP message was received.

During the dialog with the File Access Server on the remote node (which was necessary to execute your request), a message was received of the wrong type.

Action: For an explanation of the DAP message, refer to the *DECnet-DOS Programmer's Reference Manual*. For further assistance, see the person responsible for your network.

Internal error.

An internal error message was detected.

Action: See the person responsible for your network for assistance.

The remote server's buffer is too small.

During the dialog with the File Access Server on the remote node, that was necessary to execute your request, a message was received which included the maximum possible buffer size for the remaining messages. This size is too small to support the communication.

Action: See the person responsible for your network for assistance.

The remote server could not find that file.

The NDISK file that you specified was not found at that remote node.

Action: You must specify a file that exists.

The remote server reports DAP error x:x.

The remote File Access Listener is relaying a DAP error message to you. The error message consists of a pair of DAP error codes. The first error code indicates a specific error type. The second error code identifies the specific reason for that error.

Action: To determine the meanings of the displayed error codes, refer to the *DECnet-DOS Programmer's Reference Manual*. For further assistance, see the person responsible for your network.

The specified file is not a virtual disk file.

The NDISK file that you specified does not have the proper attributes. You might have made an error in the file name.

Action: Specify a file that has been created with the CREATE command.

C.1.3 Execution Error Messages

The following error messages might occur when you attempt to execute an NDU command or fail to install the virtual disk driver beforehand.

ALLOCATION used for CREATE only, value IGNORED.

The ALLOCATION parameter was not accepted with the command you specified.

Action: The ALLOCATION parameter can only be supplied with the CREATE command. You cannot use it with other NDU commands.

All the drives are in use, cannot OPEN another drive.

All four disk drives are in use.

Action: Close one of the open drives before opening another.

All the drives are in use, CREATE function not done.

You cannot create a new virtual disk because all drives are in use.

Action: Close one of the drives.

Cannot CLOSE a drive, the disk driver is not installed.

You cannot close a virtual disk without having the virtual disk driver installed.

Action: Refer to the appropriate installation guide for your system for instructions on installing the virtual disk driver. Install the virtual disk driver, and run the NDU utility to close the drive.

Cannot CREATE a drive, the disk driver is not installed.

You cannot create a virtual disk without having the virtual disk driver installed.

Action: Refer to the appropriate installation guide for your system for instructions on installing the virtual disk driver. Install the virtual disk driver, and run the NDU utility to create the disk drive.

Cannot CREATE a disk file without including NDISK parameter.

You cannot issue a CREATE command without the NDISK parameter.

Action: Reissue the CREATE command, and specify the name of the file that will contain the virtual disk on the remote node.

Cannot CREATE a disk file without including NODE parameter.

You cannot issue a CREATE command without the NODE parameter.

Action: Reissue the CREATE command, and specify the name of the remote node with NODE.

Cannot DELETE a disk file without including NDISK parameter.

You cannot issue a DELETE command without the NDISK parameter.

Action: Reissue the DELETE command, and specify the name of the file that will contain the virtual disk on the remote node.

Cannot DELETE a disk file without including NODE parameter.

You cannot issue a DELETE command without the NODE parameter.

Action: Reissue the DELETE command, and specify the name of the remote node with NODE.

Cannot DELETE, the disk driver is not installed.

You cannot delete a virtual disk without having the virtual disk driver installed.

Action: Refer to the appropriate installation guide for your system for instructions on installing the virtual disk driver. Install the virtual disk driver, and run the NDU utility to delete the disk drive.

Cannot OPEN a drive, the disk driver is not installed.

You have not installed NDU properly.

Action: Refer to the appropriate installation guide for your system for instructions on installing the virtual disk driver. Repeat the procedure, and run the NDU utility.

Cannot OPEN a drive without including NDISK parameter.

You cannot issue an OPEN command without the NDISK parameter.

Action: Reissue the OPEN command, and specify the name of the file that will contain the virtual disk on the remote node.

Cannot OPEN a drive without including NODE parameter.

You cannot issue an OPEN command without the NODE parameter.

Action: Reissue the OPEN command, and specify the name of the remote node with NODE.

Drive *name* is already OPEN; CREATE function not done.

The drive is already open. You cannot format the new data file.

Action: Use the SHOW STATUS command to list the open drives.

Drive *name* is already OPEN, cannot OPEN it again.

The drive is already open.

Action: You cannot open a drive that is already open.

Drive *name* is not a network disk.

The drive name that you specified is invalid.

Action: The drive name consists of one alphabetic character. Use the SHOW STATUS command to list the valid drive names.

Drive *name* is not a network disk, cannot CLOSE it.

You cannot close a drive which is not a virtual disk drive.

Action: Use the SHOW STATUS command to list the open drives.

Drive *name* is not a network disk, cannot OPEN it.

You can use only the drive names listed by the SHOW STATUS command.

Action: Use a proper drive name.

Drive *name* is not a network disk, CREATE function not done.

You can use only the drive names listed by the SHOW STATUS command.

Action: Use a proper drive name.

Drive *name* is not OPEN, cannot CLOSE it.

You cannot close an unopened drive.

Action: Use the SHOW STATUS command to list the status of the drives.

Drive *name* is not OPEN, cannot DELETE its disk.

You cannot delete a disk without having the drive opened.

Action: Use the SHOW STATUS command to list the status of the drives.

Drive *name* is too long. It can only be a single character.

You supplied an invalid name for the disk drive.

Action: A valid drive name consists of one alphabetic character. Use the SHOW STATUS command to list the valid drive names.

DRIVE, NDISK, or PRINTER must be specified, the CLOSE function was not done.

You cannot issue the CLOSE function without specifying DRIVE, NDISK, or PRINTER as a parameter.

Action: To perform the CLOSE function, you must specify either the name of the virtual disk drive or the file that will be the virtual disk on the remote node.

Length of NDISK parameter is too long.

File name must be less than 127 characters.

An invalid length was supplied for the file name.

Action: The NDISK parameter specifies the name of the file which will be the virtual disk on the remote node. The valid range for the file name is 1 to 126 alphanumeric characters. Enter a valid file name.

Length of NODE parameter is too long.

Node name must be less than 6 characters.

An invalid node name was supplied.

Action: A valid node name consists of one to six alphanumeric characters. It must contain at least one alphabetic character. Enter a valid node name.

Length of USER parameter is too long.

The string must be less than 120 characters.

You supplied an invalid user name.

Action: NDU could not recognize the user name. A valid user name consists of access-control information: *user-id*, *password*, and *account*. Each field can include 1 to 39 alphanumeric characters. Enter a valid USER name.

NDISK not found, the CLOSE function was not done.

Either DRIVE or NDISK must be specified; the CLOSE function was not done.

Action: To determine which drive is to be closed, you must specify a drive name, an NDISK file specification, or both.

The Disk Driver is not Installed.

You cannot create or use a virtual disk without first installing the Virtual Disk Driver.

Action: Refer to the appropriate installation guide for your system for instructions on installing the virtual disk driver. Repeat the installation procedure, and then try to run NDU.

C.2 Virtual Printer Error Messages

The following error messages might occur when you use the virtual printer feature.

C.2.1 Command Line Error Messages

The error messages listed in this section are all related to the syntax of a command line.

DELETE function failed: Connect failed, Unrecognized node name.

A valid node name consists of 1 to 6 alphanumeric characters. It must contain at least one alphabetic character. The node name can also be a unique numeric address in the format of *area.number*.

OPEN/CREATE function failed: Connect failed, Unrecognized node name.

A valid node name consists of 1 to 6 alphanumeric characters. It must contain at least one alphabetic character. The node name can also be a unique numeric address in the format of *area.number*.

Unexpected syntax error.

You mistyped a command.

Unrecognized command.

You mistyped an NDU command or tried to use a command that NDU could not recognize.

Unrecognized HELP option; try typing just HELP.

You typed an invalid HELP option. Just type HELP for assistance.

Unrecognized NPRINT file name.

A valid file name consists of 1 to 126 alphanumeric characters. The file name may include device and directory fields, but it must be in the proper format for the remote node's system type.

Unrecognized Parameter.

You supplied an invalid parameter with an NDU command. Type HELP for assistance.

Unrecognized SHOW option.

NDU could not recognize the option. The only valid option for the SHOW command is STATUS.

Unrecognized USER name.

NDU could not recognize the USER name. A valid USER name consists of access control information: *user-id*, *password*, and *account*. Each field can include 1 to 39 alphanumeric characters. Enter a valid USER name.

C.2.2 Network Errors

A network error consists of a pair of messages separated by a colon. The message format is:

ndu-specific-message : dap-specific-message

The first message can only be:

OPEN/CREATE function failed.

The second half of the message string can be one of the following DAP messages:

A received DAP message was poorly formed.

During the dialog with the File Access Server on the remote node (which was necessary to execute your request), a message was received of the proper type, but it did not contain the proper contents.

Action: For an explanation of the DAP message, refer to the *DECnet-DOS Programmer's Reference Manual*. For additional assistance, see the person responsible for your network.

An unexpected DAP message was received.

During the dialog with the File Access Server on the remote node (which was necessary to execute your request), a message was received of the wrong type.

Action: For an explanation of the DAP message, refer to the *DECnet-DOS Programmer's Reference Manual*. For further assistance, see the person responsible for your network.

Internal error.

An internal error message was detected.

Action: See the person responsible for your network for assistance.

The remote server's buffer is too small.

During the dialog with the File Access Server on the remote node (which was necessary to execute your request), a message was received which included the maximum possible buffer size for the remaining messages. This size is too small to support the communication.

Action: See the person responsible for your network for assistance.

The remote server could not find that file.

The NDISK file that you specified was not found at that remote node.

Action: You must specify a file which exists.

The remote server reports DAP error x:x.

The remote File Access Listener is relaying a DAP error message to you. The error message consists of a pair of DAP error codes. The first error code indicates a specific error type. The second error code identifies the specific reason for that error.

Action: To determine the meanings of the displayed error codes, refer to the *DECnet-DOS Programmer's Reference Manual*. For further assistance, see the person responsible for your network.

C.2.3 Execution Error Messages

The following error messages might occur when you attempt to execute an NDU command or fail to install the virtual printer driver beforehand.

Cannot CLOSE the printer, the Printer Driver is not installed.

You cannot close the printer without having the virtual printer driver installed.

Action: Refer to the installation guide for instructions on installing the virtual printer driver. Install the virtual printer driver, and run the NDU utility to close the printer.

Cannot OPEN/CREATE the printer, the Printer Driver is not installed.

You cannot open/create the virtual printer without having the virtual printer driver installed.

Action: Refer to the *DECnet-DOS Installation Guide* for instructions on installing the virtual printer driver. Install the virtual printer driver, and run the NDU utility to open/create the printer.

Cannot OPEN/CREATE the printer without including NODE parameter.

You cannot issue the OPEN or CREATE command without the NODE parameter.

Action: Reissue the OPEN or CREATE command, and specify the name of the remote node with NODE.

**Length of NODE parameter is too long.
Node name must be less than 6 characters.**

You supplied an invalid node name.

Action: A valid node name consists of one to six alphanumeric characters. It must contain at least one alphabetic character. Enter a valid node name.

**Length of NPRINT parameter is too long.
File name must be less than 127 characters.**

You supplied an invalid file name for the NPRINT parameter.

Action: A valid file name consists of 1 to 126 alphanumeric characters. The file name can include device and directory fields, but it must be in the proper format for the remote node's system type. Enter a valid file name.

**Length of USER parameter is too long.
The string must be less than 120 characters.**

You supplied an invalid user name.

Action: NDU could not recognize the user name. A valid user name consists of access-control information: *user-id*, *password*, and *account*. Each field can include 1 to 39 alphanumeric characters. Enter a valid USER name.

Printer is not OPEN, cannot CLOSE it.

You cannot close an unopened printer.

Action: Use the SHOW STATUS command to list the status of the printer.

The Printer Driver is not installed.

You cannot create or use a virtual printer without first installing the Virtual Printer Driver.

Action: Refer to the appropriate installation guide for your system for instructions on installing the virtual printer driver. Repeat the installation procedure, and then try to run NDU.

Using FAL in a Windows Environment

The File Access Listener (FAL) is available as a fully functional windows application. FAL provides a way for you to let other nodes access files on your node. FAL listens for and receives remote access requests from other nodes on the network.

This appendix describes how to use FAL in a windows environment. It does not explain how to use MS-Windows. This version of FAL works on MS-Windows V2.03 and V2.1 only. For more information about MS-Windows, refer to your MS-Windows documentation.

D.1 Starting FAL

You start FAL by selecting FAL.EXE from your \DECNET directory. To do this, use the mouse to click twice on the file name, FAL.EXE.

FAL appears as a small window on your screen. FAL continues to listen for and receive remote access requests from other nodes in the network until you close the utility.

You can also use FAL as an icon. Once you start the utility, you can change it to an icon, and FAL continues to listen for and receive connect requests and file input and output. This allows you to perform other tasks while FAL listens in the background. When you recall FAL from the icon, the utility lists all of the connect requests and file I/O it has received until that point.

D.1.1 FAL Menu Options

FAL, in a windows environment, provides two options from which you can choose. The options appear on a menu bar at the top of the FAL window. The options include the following:

- Options
- Help

You select an option by moving the mouse and clicking on the option.

D.1.1.1 Options

When you select Options, another small pop-up window appears displaying five more choices.

- ASCII
- Binary
- Do not supersede
- Log
- Trace

When you select the ASCII or the Binary options, the window disappears. FAL now transfers all files as the specified type (ASCII or Binary).

When you select the Do not supersede option, the window disappears and other users can read the files on your system, but they cannot supersede your files.

When you select the Log option, another small pop-up window appears. In this window you must specify a file specification to log all status messages in or choose to have the messages displayed in the window. If you do not indicate a file specification and you choose the file option, FAL sends messages to the default file FAL.LOG.

Once you choose the Log option, FAL logs the type of access for every request it receives in the file or in the window. The type of access includes the following information:

- The command or request (such as an NFT DIRECTORY or DELETE command).
- The requested directory name and file name.

- The node name.
- The access control information for that node (user name and account).

When you select the Trace option, another small pop-up window appears. This option starts a protocol trace. (This is generally used by system programmers.) In this window, you must specify a file to send the trace output to or choose to have the information displayed in the FAL window. If you do not indicate a file specification and you choose the file option, FAL send the output to the default file FAL.TRA.

The options provide the same functionality in windows as they do when used from the command line. The options listed here are also described (as switches) in Chapter 3.

D.1.2 Help

If you select the Help option, FAL displays a list of the available options and a brief description of each. If you need more detailed information about the FAL options, refer to Chapter 3.

1870

...

...

...

...

Using NFT in a Windows Environment

The Network File Transfer (NFT) utility is available as a fully functional windows application. NFT provides a way for you to share files with other nodes on the DECnetTM network. It allows you to list directories, copy files, append files, delete files, display files, print text files, and submit batch jobs.

This appendix tells how to use NFT in a windows environment. This version of NFT works only on MS-Windows V2.03 and V2.1. If you are not familiar with windows, refer to the MS-Window documentation.

E.1 Starting NFT

You start NFT by selecting it from your \DECNET directory. To do this, use the mouse to click twice on the file name, NFT.EXE. (You can also select RUN from the FILE menu in the MS-DOSTM Executive, and enter NFT.EXE in the pop-up window.)

NFT appears as a window on your screen. If you were using the MS-DOS Executive, the current directory of files also appears on your screen.

E.2 NFT Menu Options

To perform NFT functions, you can choose from several options. The options appear on the menu bar at the top of the NFT window, and include the following:

- Copy
- View
- ChangeDirectory

- Special
- DataType
- CharacterSet
- ShowDefaults
- Help
- Cancel ^C

Select options by moving the mouse and clicking on the option. If the option includes an underlined letter, such as the "D" in the CHANGEDIRECTORY option, you can hold down the **[Alt]** key and press the underlined letter to choose that option instead of using the mouse. For example, to choose the CHANGEDIRECTORY option, hold down the **[Alt]** key and press **D**.

E.2.1 Copy

When you select the Copy option, another small pop-up window appears displaying three sub-options.

- Copy
- Append
- Copy to Clipboard

When you select Copy or Append, a pop-up window appears requesting a source file specification, a destination file specification, the maximum record size, and allocation. You also have a choice of different attributes, character sets, and special options to use with your file. The source file specification and the destination file specification are the only fields required. If you do not specify any attributes, character sets, or special options, NFT uses the defaults.

Once you enter a source file name, a destination file specification, and choose any attributes, character sets, or special options, you can click on "OK" and have NFT start the file transfer.

NFT displays a message indicating that the file is being copied. For example, if you copied the file TEST.TXT from user SMITH on the node BLD10, you would receive the following message:

```
Copying file: BLD10::DSK1:[SMITH]TEST.TXT;1 --> C:\DECNET\TEST.TXT;1
[10230 bytes at 8412 bytes/second]
```


If you select Copy to Clipboard, NFT displays a window that requires you to enter a file specification. You can then click on "OK" to have NFT place that file in the clipboard. For information on the MS-Windows clipboard, refer to the MS-Window documentation.

E.2.2 View

The View option provides several ways of looking at a file. This option functions like the DIRECTORY command. When you select the View option, another small pop-up window appears displaying the following four options:

- Brief
- Verbose
- Get full info
- Text

The Brief option causes NFT to display a list of only the file names for the current directory.

The Verbose option causes NFT to display a list of the files names and the dates that the files were created for the current directory.

The Get full info option causes NFT to create a pop-up window that requests a file specification. When you specify the file and click on "OK," the window displays all additional information for the requested file.

The Text option causes NFT to create a pop-up window that requests a file specification. When you specify the file and click on "OK," the window displays the contents of the specified file(s).

E.2.3 Change Directory

The Change Directory option lets you select a different directory to use for choosing source or destination files. You can select a new local directory, or you can select a remote directory.

When you select this option, NFT displays a pop-up window that requests a file specification for the new directory. You can click on "OK" after you enter the file specification to change the directory, or click on "Cancel" to keep the current directory and return to the NFT window.

E.2.4 Special

The Special option provides three different operations to perform on your source file. When you select the Special option, another small pop-up window appears, displaying the following operations:

- Delete
- Print
- Submit

The Delete operation deletes the specified source file.

The Print operation applies to remote files only. You can print a remote source file using this operation.

The Submit operation applies to remote nodes only. You can submit the source file as a batch job using this operation.

When you select one of these operations, a pop-up window appears. The window requests a file specification for the requested operation. After you enter the file specification, you can click on "OK" to complete the operation or on "Cancel" to return to the NFT window.

E.2.5 Data Type

The Data Type option lets you specify the type of data that the source file contains. When you select the Data Type option, another small pop-up window appears, displaying a list of data types. The data types include the following:

- ASCII
- Image
- Block mode
- Macy11
- Line numbers

You can click on the appropriate data type. Note that if you do not select a data type, NFT tries to determine the type by default. Some binary files may appear to NFT as ASCII files, and they are not transferred properly. In order to avoid this, you should select the "Image" data type for use with binary files.

E.2.6 Character Set

If you want to use a character set translation while running NFT in a windows environment, you must use the *.CHR files. These files contain all of character set information you need. The files should already exist in your DECnet directory if you used the DECnet PCSA Client for DOS installation Procedure. If the files are not in your directory, refer to the appropriate installation guide.

The Character Set option on the main menu lets you select the type of character set to use for the source file. This option is applicable only for ASCII files. It is necessary to change character sets only if the source and destination file character sets differ from each other.

When you select the Character Set option, another small pop-up window appears displaying the available character sets. The character sets you can select from include the following:

- No translation
- DEC Multinational
- USA
- Dutch
- Finnish
- French-Canadian
- French
- German
- ISO Latin-1
- Italian
- Norwegian/Danish
- Portuguese
- Spanish
- Swedish
- Swiss
- U.K.

If you do not choose a character set, NFT uses the "No translation" option as the default.

When copying files, another character set option appears in the copy menu. This character set option applies only to the destination file, and it is necessary only if the source file and the destination file use different character sets. For example, to convert a file on a remote system to the character set specified for your personal computer (not the character set used inside windows), specify a character set for the source file only. When converting a file from your personal computer to another character set on a remote system, specify a character set for the destination file only.

NOTE

The default character set used in the windows environment is ISO Latin-1.

E.2.7 Show Defaults

The Show Defaults option displays the temporary default table of remote access information. The password is not displayed on the screen. When you choose this option, a pop-up window appears displaying the access information.

If you defined remote access for BLD10"J_SMITH OPEN"::, for example, the Show Defaults window contains the following:

```
BLD10"J_SMITH"::
```

Notice that the password is omitted from the information displayed. If you have not defined any remote access information, the Show Defaults window is empty.

E.2.8 Help

The Help option displays brief information about NFT and lets you choose other options from the menu bar to get information about.

As you select other options, the information in the Help pop-up window changes to reflect the option specified. For example, if you select the Copy option immediately after the Help option, the information in the Help window refers to the Copy option.

E.2.9 Cancel (^C)

The Cancel (^C) options aborts any file transfers that are in process. When you invoke NFT, this option appears shaded. The shading means that the option is not available. As soon as you start to transfer a file, the option becomes available for use.

Running Asynchronous DECnet Communications

To run asynchronous DECnet™ communications (Async DECnet), you must configure your personal computer for asynchronous Digital Data Communications Message Protocol (DDCMP™). For more information on configuring your personal computer as an asynchronous DDCMP node, refer to *Installing DECnet PCSA Client for DOS (with Diskettes)*.

After you install the necessary software, you can use specific SETHOST scripts that contain the commands needed to run Async DECnet on your personal computer. You must customize these scripts to support your particular configuration.

NOTE

In an Asynchronous DECnet environment, you must issue an NCP SET LINE STATE OFF command before you unload the network. If you do not do this, you will not be able to reload the network without rebooting your personal computer.

F.1 Associated Documents

The following list provides documentation that you can refer to for detailed information on specific topics:

For information on	Refer to
DECnet Software	<i>Installing DECnet PCSA Client for DOS (with Diskettes)</i>
Making and testing DECnet connections	<i>Installing DECnet PCSA Client for DOS (with Diskettes)</i> and the <i>DECnet-DOS Network Management Guide</i>
Setting up a router	The <i>DECnet-VAX Networking Troubleshooting Guide</i>
The Network Control Program	The <i>DECnet-DOS Network Management Guide</i>
DYNSWITCH	The <i>DECnet-VAX Networking Guide</i> and the <i>DECnet/PCSA Release Notes</i>
DECnet-DOS and the SETHOST Utility	The <i>DECnet-DOS User's Guide</i> and the <i>DECnet-DOS SETHOST Terminal Emulation Guide</i>
Communications settings for your modem	The <i>DECnet-DOS User's Guide</i>

F.2 Running Asynchronous DECnet

You can run Async DECnet using any of the following combinations:

- A hardwired dedicated DECnet line
- A hardwired dynamically switched terminal line
- A modem with a dedicated DECnet line
- A modem with a dynamically switched terminal line

Dedicated DECnet lines are permanently configured as DECnet lines. They can connect your personal computer to another personal computer (Async DECnet), a DECrouter™, or a VMS™ system.

Dynamically switched terminal lines must be switched from terminal to DECnet DDCMP lines for use with DECnet communications. They connect your personal computer to VMS systems running DECnet. The lines are switched for the duration of the dialup connection only. The remote VMS node must have the asynchronous DDCMP driver, NODEVICE loaded and the image, DYNSWITCH installed. You must have system manager privileges to load the NODEVICE driver and install DYNSWITCH.

NOTE

You cannot use DYNSWITCH to switch an asynchronous terminal connection to a VMS

system, through a switch or a LAT terminal server, to a DDCMP line. DYN SWITCH consumes too many terminal server buffers.

For more information on DYN SWITCH, refer to the *DECnet-VAX Networking Guide*.

F.3 Using the SETHOST Scripts

A script is a text file containing commands that allow the SETHOST utility to perform many operations automatically. Scripts are useful when running asynchronous DECnet communications because you can use them to automate the asynchronous connection procedure. Otherwise, you would have to enter a series of commands each time you want an asynchronous connection.

If you used the DECnet PCSA Client for DOS installation procedure, the scripts are in your \DECNET directory. If not, you can copy them manually. DECnet PCSA Client for DOS provides the following three scripts:

Script	Use With
HARDDYN.SCR	A hardwired dynamically switched terminal line
DIALNET.SCR	A modem with a dedicated DECnet line
DIALDYN.SCR	A modem with a dynamically switched terminal line

Note: No script is needed for hardwired dedicated DECnet lines.

Once you choose the appropriate script, edit it to supply the information needed to make a successful connection.

After you edit the appropriate script, specify it when you start SETHOST. For example, if you want to run the DIALNET.SCR script, enter the following:

```
SETHOST /SCRIPT=DIALNET
```

DIALNET specifies the name of the script file that you want to run. The default file type for script files is .SCR, so, you do not have to specify the type.

If you are not familiar with the SETHOST utility, refer to the *DECnet-DOS SETHOST Terminal Emulation Guide* for more information.

F.3.1 Script Commands

To run Async DECnet, you might need to use the provided SETHOST scripts. This section describes the script commands that you need to use when editing the scripts. The following commands are used in the scripts:

- BAUD RATE
- DIAL
- PORT

BAUD RATE

The BAUD RATE command sets the rate at which characters are both received and transmitted.

Note: If you are dialing a router, you must have the same baud rate as the router. Routers cannot automatically determine the line speed on a dial-up line.

Format

BAUD RATE: *speed*

Remarks

The BAUD RATE command sets the rate at which characters are both received and transmitted.

The value of *speed* must be one of the following:

50	300	2400
75	600	3600
110	1200	4800
134	1800	9600
150	2000	19200

The BAUD RATE command is invalid when you use NETWORK, LAT, or CTERM in a PORT command.

The RX BAUD and TX BAUD commands are exactly the same as this command.

See the PORT command.

Example Coding

```
PORT: Data-1
PARITY: None
DATA BITS: 8
STOP BITS: 1
BAUD RATE: 2400
DIAL:B
```

Results

- 0 - Command Successfully Completed
- 4 - Invalid Parameter
- 5 - Incompatible Settings

DIAL

The DIAL command turns on the DTR signal and dials a phone number.

Note: If you are dialing up to a router, you must be in the same area as the router to connect.

Format

DIAL: *dial-string*

Remarks

The DIAL command turns on the DTR signal between the computer and the modem, then sends a *dial-string* to the modem.

The DIAL command "compresses" *dial-string* before sending it by removing spaces, tab characters, and hyphens (-).

The DIAL command is equivalent to a DTR SET command followed by a SEND command that sends a "compressed" *dial-string* to the modem.

If you need to send embedded spaces, tab characters, or hyphens in the *dial-string*, use the SEND command instead of DIAL.

If *dial-string* is a single letter from A to J, DIAL sends the corresponding Set Up phone number string to the modem.

The DIAL command is invalid when you use NETWORK, LAT, or CTERM in a PORT command.

See the PORT command.

Example Coding

This example shows a usual combination of communication parameters.

```
PORT: Data-1
PARITY: None
DATA BITS: 8
STOP BITS: 1
BAUD RATE: 2400
DIAL: 1 617 555-1212
```

PORT

The PORT command tells the script processor how to communicate with the host computer.

Format

PORT: *method*

where *method* is one of the following:

DATA-1
MODEM-1
DATA-2
MODEM-2
INTEGRAL-2
CTERM
LAT
NETWORK

Remarks

If there is no PORT command, the SETHOST Set-Up defaults are used.

DATA-1 means Port 1, data leads only.

MODEM-1 means Port 1, full modem control.

DATA-2 means Port 2, data leads only.

MODEM-2 means Port 2, full modem control.

INTEGRAL-2 means Port 2, integral modem.

CTERM specifies the DECnet CTERM protocol.

LAT specifies the LAT protocol.

NETWORK specifies use of the LAT protocol, but if that fails, then CTERM.

Example Coding

PORT: NETWORK

PORT: Data-1

Results

- 0 - Command Successfully Completed
- 4 - Invalid Parameter
- 19 - Comm Port not available

F.4 Using a Hardwired Dedicated DECnet Line

No script is needed to run Async DECnet on hardwired dedicated DECnet lines with asynchronous DDCMP configured personal computers. Once the DECnet-DOS™ installation procedure completes, reboot your personal computer and Async DECnet runs automatically.

To verify your DECnet connection, use the Network Control Program (NCP) LOOP EXECUTOR and LOOP NODE commands. For more information about verifying DECnet, refer to *Installing DECnet PCSA Client for DOS (with Diskettes)*.

F.5 Using a Hardwired Dynamically Switched Terminal Line

DECnet-DOS provides the HARDDYN.SCR script to use with hardwired dynamically switched terminal lines. This script contains the commands to log in, turn on DECnet, and establish an asynchronous DECnet connection.

You must review the script to make sure all settings are correct for your configuration. You may have to edit the script to change the following two settings:

- BAUD RATE:*speed*
- PORT:*method*

The BAUD RATE command sets the rate at which characters are both received and transmitted.

The PORT command tells the script processor how to communicate with the host computer.

F.5.1 The HARDDYN.SCR Script

The following script is available for hardwired dynamically switched terminal line connections.

COMMENT: This script logs in, turns DECnet on, and establishes an
COMMENT: Async DECnet connection via the PC's COM1 port.

COMMENT: Please change the username and password from "USER" to the your
COMMENT: username and password or use the SEND USERNAME [node] and
COMMENT: SEND PASSWORD [node] script commands to extract the username and
COMMENT: password from the DECnet database on your PC.

```

COMMENT: Reset the error conditions.
  ON ERROR:
  END ON ERROR:

COMMENT: Here is where we specify what port and baud rate to use.
  PORT: Data-1
  BAUD RATE: 2400

$connect:
  DISPLAY:Connected to remote system<CR><LF>
  PAUSE: 0:0:1
  SEND:<CR>
  WAIT FOR:Username:
  SEND:USER<CR>
  WAIT FOR:Password:
  SEND:USER<CR>
  PAUSE: 0:0:2
  WAIT FOR:$
  DISPLAY:Logged into remote system<CR><LF>

COMMENT:  Reset the error conditions.
  ON ERROR:
  END ON ERROR:

COMMENT: The following VMS DCL command will turn the
COMMENT: terminal line into a DECnet (DDCMP) line.
  SEND:set terminal/switch=DECnet/protocol=DDCMP/manual<CR>

COMMENT: VMS will time out the attempt if it sees no response in
COMMENT: 4 minutes (2400 seconds), so there is no need to wait
COMMENT: any longer than that.
  TIMER: 2400
  ON ERROR:
    GOTO: $Other end did not start DECnet
  END ON ERROR:

COMMENT: Watch for the response from VMS.
COMMENT: If we do not see it, we go to the above ON ERROR segment.
  WAIT FOR:line

COMMENT: If we get here, we must have received the correct response from VMS.
COMMENT: So, we use NCP to turn the line state on.
COMMENT: If the NCP command fails, we assume its because DECnet
COMMENT: is not installed, and tell the user.
  ON ERROR:
    GOTO: $Install DECnet and retry setting line state on
  END ON ERROR:
  SYSTEM:ncp set line state on
  EXIT EMULATOR:

```



```

$Other end did not start DECnet:
  DISPLAY:Other end did not start DECnet in time.<CR><LF>
  DISPLAY:Would you like to hang up and try again?
  READ: answer
  CASE: answer
    "Y" GOTO: $retry
    "y" GOTO: $retry
  DEFAULT: EXIT EMULATOR:
  CASE END:

$retry:
COMMENT: Hang up the phone and start over.
COMMENT: Leave DTR off for 4 seconds to make sure
COMMENT: it really hung up.
  DTR CLEAR:
  PAUSE: 0:0:4
  DTR SET:
  GOTO: $connect

$Install DECnet and retry setting line state on:
COMMENT: Tell the user to do it, because a script can't install DECnet
COMMENT: without leaving a hole in memory when SETHOST exits.

  DISPLAY>Please install DECnet and retry setting the line state on.<CR><LF>
  EXIT EMULATOR:

```

F.6 Using a Modem with a Dedicated DECnet Line

DECnet-DOS provides the DIALNET.SCR script to use for modems with dedicated DECnet lines. This script contains the commands to dial up an asynchronous DECnet line and establish an asynchronous DECnet connection.

You must review the script to make sure that all settings are correct for your configuration. You may need to edit the following three settings:

- BAUD RATE:*speed*
- DIAL:*dial-string*
- PORT:*method*

The BAUD RATE command sets the rate at which characters are both received and transmitted.

The DIAL command turns on the Data Terminal Ready (DTR) signal between the computer and the modem, then sends a *dial-string* to the modem.

The PORT command tells the script processor how to communicate with the host computer.

F.6.1 The DIALNET.SCR Script

The following script is available for dedicated DECnet line connections using a modem. (Hayes® is a registered trademark of Hayes Microcomputer Products, Inc.)

```
COMMENT: This script dials up an Async DECnet line (i.e. DECrouter 200),
COMMENT: and establishes an Async DECnet connection via the PC's COM1 port.

COMMENT: There are two types of modem command languages in this script:
COMMENT: DMCL (Digital Modem Command Language) offered by Digital modems and
COMMENT: Hayes-compatible (AT Modem Command Language).
COMMENT: Since both can't be used at the same time the Hayes-compatible
COMMENT: script commands are commented out.
COMMENT: Please change the telephone number below before using this script.

COMMENT: Reset the error conditions.
    ON ERROR:
    END ON ERROR:

COMMENT: Tell DECnet to stop using the comm port so SETHOST can use it.
    SYSTEM:ncpset set line state hangup

COMMENT: Here is where we specify what port and baud rate to use.
    PORT: Data-1
    BAUD RATE: 2400

$dialit:
COMMENT: Dial the modem.
    NO XON/XOFF:
    TIMER:1
    ON ERROR:
    GOTO: $dialit
    END ON ERROR:

    COMMENT: Tone dial a Scholar Plus modem in
    COMMENT: DMCL (Digital Modem Command Language) mode.
    DIAL:<CTRL/B>
    WAIT FOR:Ready
    SEND: dial t15551212<cr>

    COMMENT: Tone dial a Hayes-compatible
    COMMENT: (AT Modem Command Language) modem.
COMMENT: DIAL:AT<CR>
COMMENT: WAIT FOR:OK
COMMENT: SEND:ATDT12345678<CR>
```

```

ON ERROR:
    DISPLAY: <CTRL/G>Other end did not answer!<CR><LF>
    PAUSE: 0:0:5
    EXIT EMULATOR:
END ON ERROR:

TIMER:60
    COMMENT: Wait for "CONNECT" if Hayes-compatible is used.
    COMMENT: WAIT FOR:CONNECT
    COMMENT: Or wait for "Attached" if DMCL is used.
    WAIT FOR:Attached
    DISPLAY:Connected to remote system<CR><LF>
    PAUSE: 0:0:1

COMMENT:    Reset the error conditions.
ON ERROR:
    END ON ERROR:

COMMENT: Use NCP to turn the line state on.
COMMENT: If the NCP command fails, we assume its because DECnet
COMMENT: is not installed, and tell the user.
    ON ERROR:
        GOTO: $Install DECnet and retry setting line state on
    END ON ERROR:
    SYSTEM:ncp set line state on
    EXIT EMULATOR:

$Other end did not start DECnet:
    DISPLAY:Other end did not start DECnet in time.<CR><LF>
    DISPLAY:Would you like to hang up and try again?
    READ: answer
    CASE: answer
        "Y" GOTO: $retry
        "y" GOTO: $retry
    DEFAULT: EXIT EMULATOR:
    CASE END:

$retry:
COMMENT: Hang up the phone and start over.
COMMENT: Leave DTR off for 4 seconds to make sure
COMMENT: it really hung up.
    DTR CLEAR:
    PAUSE: 0:0:4
    DTR SET:
    GOTO: $dialit

$Install DECnet and retry setting line state on:
COMMENT: Tell the user to do it, because a script can't install DECnet
COMMENT: without leaving a hole in memory when SETHOST exits.

    DISPLAY>Please install DECnet and retry setting the line state on.<CR><LF>
    EXIT EMULATOR:

```


F.7 Using a Modem with a Dynamically Switched Terminal Line

DECnet-DOS provides the DIALDYN.SCR script to use for modems with dynamically switched terminal lines. This script dials up a computer, turns DECnet on, and establishes an asynchronous DECnet connection.

You must review the script to make sure that all settings are correct for your configuration. You may need to edit the following three settings:

- BAUD RATE:*speed*
- DIAL:*dial-string*
- PORT:*method*

The BAUD RATE command sets the rate at which characters are both received and transmitted.

The DIAL command turns on the Data Terminal Ready (DTR) signal between the computer and the modem, then sends a *dial-string* to the modem.

The PORT command tells the script processor how to communicate with the host computer.

F.7.1 The DIALDYN.SCR Script

The following script is available for use with dynamically switched terminal line connections using modems.

```
COMMENT: This script dials up a computer, logs in, turns DECnet on, and
COMMENT: and establishes an Async DECnet connection via the PC's COM1 port.

COMMENT: There are two types of modem command languages in this script:
COMMENT: DMCL (Digital Modem Command Language) offered by Digital modems and
COMMENT: Hayes-compatible (AT Modem Command Language).
COMMENT: Since both can't be used at the same time the Hayes-compatible
COMMENT: script commands are commented out.
COMMENT: Please change the telephone number below before using this script.
COMMENT: Please change the username and password from "USER" to the your
COMMENT: username and password or use the SEND USERNAME [node] and
COMMENT: SEND PASSWORD [node] script commands to extract the username and
COMMENT: password from the DECnet database on your PC.

COMMENT: Reset the error conditions.
ON ERROR:
END ON ERROR:

COMMENT: Tell DECnet to stop using the comm port so SETHOST can use it.
SYSTEM:ncpset line state hangup
```


COMMENT: Here is where we specify what port and baud rate to use.

PORT: Data-1

BAUD RATE: 2400

\$dialit:

COMMENT: Dial the modem.

NO XON/XOFF:

TIMER:1

ON ERROR:

GOTO: \$dialit

END ON ERROR:

COMMENT: Tone dial a Scholar Plus modem in

COMMENT: DMCL (Digital Modem Command Language) mode.

DIAL:<CTRL/B>

WAIT FOR:Ready

SEND: dial t15551212<cr>

COMMENT: Tone dial a Hayes-compatible

COMMENT: (AT Modem Command Language) modem.

COMMENT: DIAL:AT<CR>

COMMENT: WAIT FOR:OK

COMMENT: SEND:ATDT12345678<CR>

ON ERROR:

DISPLAY: <CTRL/G>Other end did not answer!<CR><LF>

PAUSE: 0:0:5

EXIT EMULATOR:

END ON ERROR:

TIMER:60

COMMENT: Wait for "CONNECT" if Hayes-compatible is used.

COMMENT: WAIT FOR:CONNECT

COMMENT: Or wait for "Attached" if DMCL is used.

WAIT FOR:Attached

DISPLAY:Connected to remote system<CR><LF>

PAUSE: 0:0:1

SEND:<CR>

WAIT FOR:Username:

SEND:USER<CR>

WAIT FOR:Password:

SEND:USER<CR>

PAUSE: 0:0:2

WAIT FOR:\$

DISPLAY:Logged into remote system<CR><LF>

COMMENT: Reset the error conditions.

ON ERROR:

END ON ERROR:

COMMENT: The following VMS DCL command will turn the

COMMENT: terminal line into a DECnet (DDCMP) line.

SEND:set terminal/switch=DECnet/protocol=DDCMP/manual<CR>

```

COMMENT: VMS will time out the attempt if it sees no response in
COMMENT: 4 minutes (2400 seconds), so there is no need to wait
COMMENT: any longer than that.
    TIMER: 2400
    ON ERROR:
        GOTO: $Other end did not start DECnet
    END ON ERROR:

COMMENT: Watch for the response from VMS.
COMMENT: If we do not see it, we go to the above ON ERROR segment.
    WAIT FOR: line

COMMENT: If we get here, we must have received the correct response from VMS.
COMMENT: So, we use NCP to turn the line state on.
COMMENT: If the NCP command fails, we assume its because DECnet
COMMENT: is not installed, and tell the user.
    ON ERROR:
        GOTO: $Install DECnet and retry setting line state on
    END ON ERROR:
    SYSTEM: ncp set line state on
    EXIT EMULATOR:

$Other end did not start DECnet:
    DISPLAY: Other end did not start DECnet in time.<CR><LF>
    DISPLAY: Would you like to hang up and try again?
    READ: answer
    CASE: answer
        "Y" GOTO: $retry
        "y" GOTO: $retry
    DEFAULT: EXIT EMULATOR:
    CASE END:

$retry:
COMMENT: Hang up the phone and start over.
COMMENT: Leave DTR off for 4 seconds to make sure
COMMENT: it really hung up.
    DTR CLEAR:
    PAUSE: 0:0:4
    DTR SET:
    GOTO: $dialit

$Install DECnet and retry setting line state on:
COMMENT: Tell the user to do it, because a script can't install DECnet
COMMENT: without leaving a hole in memory when SETHOST exits.

    DISPLAY: Please install DECnet and retry setting the line state on.<CR><LF>
    EXIT EMULATOR:

```

Modem Control

The information in this appendix is provided for people who need to know how modem control is implemented and who understand modem operations.

G.1 Using Modem Control

To use modem control for DECnet on your personal computer, you should make sure that cables and connections conform to the EIA RS-232-C and CCITT V.28 and V.24 standards.

The following assumptions have been made for this release of DECnet-DOS:

- Connections that were established before turning the line state ON will be maintained.
- Connections that still exist when turning the line state OFF will be maintained (unless the data set is specifically commanded to hang up).

You should also be aware of the following information:

- In the United Kingdom (UK), the R1 relay in modem number 2B is held up for approximately 2 seconds in order to bridge the gaps in the ringing signal. For proper operation in calls that are automatically answered, the 2-second hold up requires that the amount of time from DSR=ON through DTR=OFF and back to DSR=ON be at least 5 seconds. This will avoid a false second call seizure. (Refer to state 7 in Figure G-1.)
- It is assumed that the terminal has two modes of operation:
 - data leads only (modem control signals ignored)

— full modem control (modem control signals acknowledged)

The data leads only mode is used for local connections, such as null modem applications. The proper mode signals are simulated internally to allow the use of a single control flow. You can choose between the two modes by using a set-up parameter.

G.2 Modem States

Figure G-1 shows the different states for the modem. (State 1 and state 2 refer to power ON and internal self-test functions, and are omitted from this diagram for purposes of clarity.)

The following sections explain the modem states that are detailed in Figure G-1.

G.2.1 General

The general states include:

- 1, OFF, or MARK — these relate to the negative voltage at the interface pin.
- 0, ON, or SPACE — these relate to the positive voltage at the interface pin.

G.2.2 State 1

This state is entered at power on.

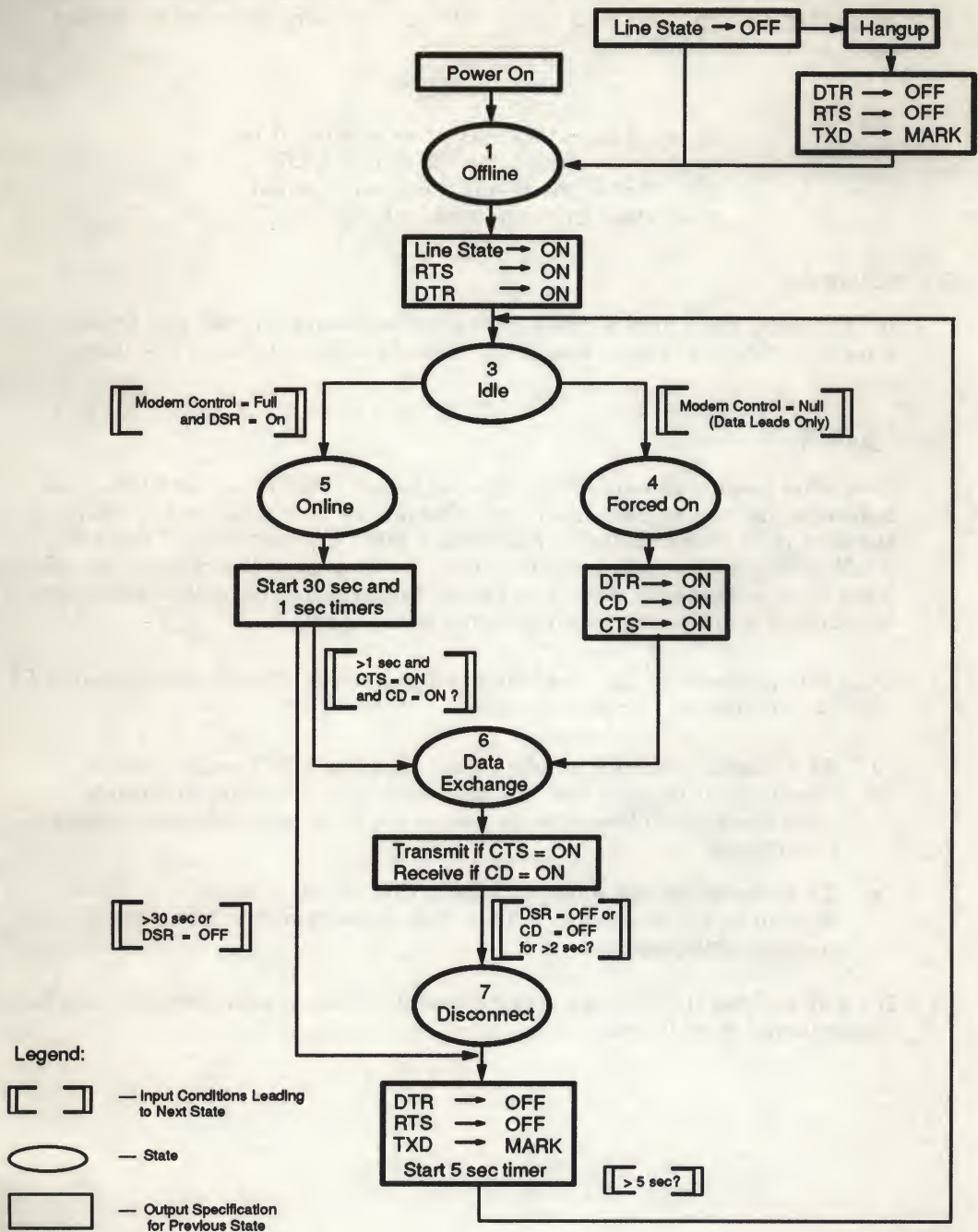
G.2.3 State 2

This state is entered if the terminal is on-line.

G.2.4 State 3

This state exits unconditionally to state 4 if the data leads only mode is selected. If the full modem control mode is selected, state 5 is entered (as soon as Data Set Ready, DSR, has come on). The wait for DSR is untimed. State 3 is the state that an on line terminal is normally in before connection has been established.

Figure G-1: Modem States



LKG-3216-891

In state 3, the Data Terminal Ready (DTR) and Request to Send (RTS) signals are turned on. In this state, the modem can switch itself on-line only if DTR is ON. The RTS signal is needed by some modems to prepare the transmit channel.

NOTE

In some cases (such as private wire), it is possible for DSR to be ON before DTR is ON. This is possible if there is a manual connection from the modem to the line.

G.2.5 State 4

In this state, the actual signals from the modem are ignored and forced internally ON (if the data leads only mode is selected). State 6 is then entered.

G.2.6 State 5

This state is entered only after Data Set Ready (DSR) has come ON. This indicates that the modem is on-line. The modem can come on-line either automatically or manually. It will come on-line *automatically* if there is a call coming in through the automatic answer unit of the modem. It will come on-line *manually* when you switch from voice to data after you have established an incoming or an outgoing manual call.

Once the modem is on-line, two timers are started. The timers are called T1 and T2. The timers operate as follows:

- T1 protects against a nondata call. This is a PTT requirement in Germany. If no data link has been established within 30 seconds after the modem has gone on-line (state 6), an automatic disconnect is initiated.
- T2 protects against data transfer within the first second after the modem has gone on-line. This avoids garbage due to transient effects in some older modems.

If Clear to Send (CTS) and Carrier Detect (CD) have both come ON, the link is considered established.










G.2.7 State 6

This state is the normal data exchange state of the terminal when the call is established.

G.2.8 State 7

This is the disconnect state. The disconnect is initiated by loss of carrier (CD) for greater than 2 seconds, or by the loss of Data Set Ready (DSR). The terminal waits 5 seconds, then a new connection is anticipated by entering state 3. This sequence provides a proper disconnection for all known modems. Figure G-2 illustrates the minimum implementation of the V.24 circuits for full duplex modem control.

Figure G-2: Minimum Implementation of V.24 Circuits for Full Duplex Modem Control

Internal DTE Signal Name (Typical)			ISO 2110 Male Connector Pin Number		V.24 Circuit Number
PROT	GND		1	Protective Ground	101
	TxD		2	Transmitted Data	103
	RxD		3	Receive Data	104
	RTS		4	Request To Send	105
	CTS		5	Ready for Sending	106
	DSR		6	Data Set Ready	107
SIG	GND		7	Signal Ground	102
	CD		8	Receive Line Carrier Detect	109
	DTR		20	Data Terminal Ready	108/2

See state 7 in Figure G-1

LKG-3217-891

G.3 Interface Requirements

The following tables specify the interface leads which are required to use DECnet-DOS in asynchronous mode, either with a direct connection or with a modem. (The DECnet software will not monitor the modem leads when direct connection is specified. This allows for the connection to a remote node using a null modem, a modem eliminator, or null modem cable. These are cables and devices which may not provide all the signal leads required for a modem connection.)

Following each table is a list of the part numbers for the Digital EIA RS-232 communication cables that will provide these leads.

Table G-1: Asynchronous Mode, Direct Connection

Pin No.	Function	CCITT V.24 Circuit No.	EIA RS-232-C Circuit No.
1	Protective Ground	101	AA
2	Transmit Data	103	BA
3	Receive Data	104	BB
7	Signal Ground	102	AB

Communication Cables:

BC22D 6	wire null modem cable (transmit and receive leads crossed within cable, eliminating the need for a separate null modem device)
BC22E 16	wire modem cable
BC22F 25	wire modem cable
BCC14 16	wire modem cable (with thumb screw connectors)
BCC04 25	wire modem cable (with thumb screw connectors)

Table G-2: Asynchronous Mode, Modem Connection

Pin No.	Function	CCITT V.24 Circuit No.	EIA RS-232-C Circuit No.
1	Protective Ground	101	AA
2	Transmit Data	103	BA
3	Receive Data	104	BB
4	Request to Send	105	CA
5	Clear to Send	106	CB
6	Data Set Ready	107	CC
7	Signal Ground	102	AB
8	Carrier Detect	109	CF
20	Data Terminal Ready	108/2	CD

Communication Cables:

BC22E 16	wire modem cable
BC22F 25	wire modem cable
BCC14 16	wire modem cable (with thumb screw connectors)
BCC04 25	wire modem cable (with thumb screw connectors)
H8571-E 25	DECconnect cable
H8571-J 9	AT Type DECconnect cable

G.4 Modem Option Selections

To use a modem with the asynchronous port, the modem must be configured to match the line characteristics used by DECnet-DOS. For many modems these will be the standard factory settings. Table G-3 lists some of the user-selectable options you may find on your modem.

Table G-3: User-Selectable Modem Options

Option	Description	Setting for Use with DECnet-DOS
Character Length	Number of bits used to encode a character.	10 bits — 8 data bits, 1 start bit, and 1 stop bit.
Speed Indicate/Select	Indicate mode: modem speed on pin 12 (CCITT V.24 112).	Indicate mode. (Pin 23, CCITT V.24 111, is never inserted, and pin 12, CCITT V.24 112, is not monitored by DECnet-DOS).
Receive Responds to RDL	Allows modem to respond to a remote digital loopback test request.	Enabled (if you want to allow a remote modem test).

Other selectable modem options should be set according to the modem user's guide for the particular modem application you are using. (Other options include a public switched line, a leased line, asynchronous or synchronous mode, and so on.)

Glossary

access-control information

Optional security information that you might need in order to access a remote node.

account

A character string consisting of 1 to 39 alphanumeric characters.

adjacent node

A node in which you are physically connected by a single line.

alphanumeric

A string of characters that contains alphabetic or numeric characters, or a combination of both.

application

A program (other than the operating system) that performs specific functions in order to meet user requirements.

buffer

A temporary storage area in a node's memory. Buffers temporarily hold data being transferred to and from the node. The size and the number of buffers determine the amount of data that can be stored.

circuit

The communications path that operates over a physical line connecting two nodes.

command switch

A word or character string that modifies the way a command operates.

communication path

The route through hardware components and a cable that a message takes when it is sent from one node to another.

counter

A counter displays statistics about the flow of network messages for your node. Counters record error conditions and accumulate their totals.

device

A specific name for a disk or diskette that is currently storing data files.

directory

A specific name assigned to a collection of files stored on disk or a diskette. A directory can provide a list of the file names that it contains.

emulate

To imitate the performance of a terminal on another system. When your personal computer emulates a terminal on a host system, it performs many of the same functions as the host terminal.

end node

A node that can send and receive information for its own use only. Your personal computer is an end node.

executor node

A node that runs Network Control Program (NCP) commands. For a personal computer, this is the local node.

file specification

A complete file identification including an optional drive name and path name. The file specification must include at least a file name followed by an optional file type and version number.

foreign file

A file that has a format not recognizable to DECnet-DOS™. A foreign file specification must be enclosed in quotation marks (" ") in a command line.

host node

A node on the network that your node can access for the purpose of sharing resources and information.

line

The physical line connecting the local node with the adjacent node.

local node

The node that you are currently using to communicate with other nodes.

logging

The process of recording information from an occurrence in the network. Logging is the process that generates a record of the event.

loopback connector

A hardware device that temporarily connects the output portion of a circuit to the input portion of a circuit for the purpose of testing communication lines. The loopback connector verifies the operation of devices on the same communications link.

loopback test

A test that sends an electronic signal over a hardware or software connection to verify the operation of devices along the communication link. Certain loopback tests require a loopback connector (see **loopback connector**).

modem

A hardware device capable of changing communications signals from digital to analog and back to digital. The term *modem* is a blend of the words *modulation* and *demodulation*.

modulation

The process of varying an original computer communication (digital) signal into a telephone communication (analog) signal.

network

A group of interconnected computers or systems that communicate with each other to share resources and information.

network coordinator

A person who assigns and updates node names and node addresses. The network coordinator provides administrative assistance to network users.

network link

A temporary connection that establishes communication between programs running on different nodes. The SETHOST utility with network virtual terminal services is a program that establishes a network link with another node, enabling you to log in to a remote host system.

node

A computer with the necessary hardware and software to participate in a network with other computers.

node address

A unique numeric character string that identifies a node to other systems in the network.

node definition

A character string that identifies a particular node. The definition includes both the node address and the node name.

node name

A one to six character alphanumeric string (containing at least one letter) that identifies a node to other users in the network.

password

A character string that uniquely confirms your identity to the system.

process name

A character string that identifies one particular user in a group of users sharing the same system.

protocols

Rules or formats that operating systems must follow to conduct effective communications with other computers in a network.

queue

A waiting line for completing a service, such as transferring files.

reachable node

A node to which the local node has a usable communications path.

remote node

Any node on the network other than the node you are currently using.

remote node list

A list containing node definitions of other nodes that are on the same network.

router

A DECnet node that can receive messages and information from one node and forward them to another node.

routing node

A DECnet node that can receive and forward information from one node to another. A routing node can perform other functions that are not limited strictly to routing information.

segment

A set of data that can be stored anywhere in memory and can be retrieved at any time.

TO *file-id*

A qualifier that specifies the name of the output file that stores information resulting from the execution of an NCP command.

topology

The physical arrangement of nodes and connecting hardware that make up the network.

user name

A character string consisting of 1 to 39 alphabetic characters identifying a user at a remote node.

window

A rectangular section of your screen used for running programs or applications.

Index

A

- Access-control information, 4-6
 - changing defaults, 4-9
 - defining for remote nodes, 4-6
 - defining with NFT, 4-7
 - displaying, 4-10
 - displaying access defaults, 4-38
 - setting defaults, 4-37
 - using defaults, 4-8
- Accessing a remote printer, 5-5
- Accessing remote files, 4-1, A-1
- Account information, 4-6
- Appending files, 4-14, 4-20
- ASCII file, 4-11, 4-12
- Asynchronous DECnet Communications
 - running, F-1

B

- Batch files
 - used with NFT, 4-17
- BAUD RATE, F-5
- Binary, 4-11

C

- Command files

- Command files (Cont.)

- submitting, 4-39
 - used with NFT, 4-2, 4-17
- Copying a local file to a remote node, 4-15
- Copying a remote file to a local node, 4-15
- Copying files, 4-10, 4-11, 4-12, 4-13, 4-14, 4-23

D

- Data

- ASCII, 4-11, 4-12
 - Binary, 4-11
 - Image, 4-11, 4-12
- DECnet-DOS
 - capabilities, 1-4
 - definitions
 - see* Glossary
 - utilities
 - FAL, 3-1, D-1
 - Job Spawner, 2-1
 - MAIL, 6-1
 - NDU, 5-1
 - NFT, 4-1, E-1
 - summary of, 1-5
- Deleting files, 4-15, 4-29

DIAL, F-7
Directories
 displaying information, 4-30

E

EXIT
 NFT, 4-33

F

FAL, 3-1, D-1
 starting
 from command line, 3-1
 from windows, D-1
File Access Listener *see* FAL
Files
 appending, 4-14, 4-20
 copying, 4-14, 4-23
 deleting, 4-15, 4-29
 displaying contents of, 4-17, 4-40
 printing, 4-16, 4-36
 types of, 4-11
File specifications, 4-10
 foreign, A-2
 local, 4-10
 naming files, 4-10
 remote, 4-10
 remote files, A-1
 using wildcards, A-2, A-3

H

Help
 MAIL, 6-9, 6-11
 NDU, 5-11, 5-18
 NFT, 4-5, 4-34

I

Image file, 4-11, 4-12
Installing virtual device programs,
 5-5

J

Job Spawner, 2-1
 creating a DECOBJ.DAT file, 2-2
 starting, 2-4

L

Local files
 copying to a remote node, 4-10,
 4-11, 4-13, 4-15
 naming, 4-10
 providing access, 3-1, 4-1
Logging file operations, 4-18

M

MAIL, 6-1
 commands
 at command line, 6-5
 at MAIL prompt, 6-6
 EXIT, 6-10
 HELP, 6-11
 MAIL, 6-12
 SEND, 6-12
 SHOW DEFAULTS, 6-14
 command summary, 6-9
 getting help for, 6-11
 HELP in, 6-9
 initialization file, 6-1
 initialization file parameters
 CARBON_COPY, 6-4
 DEFAULT_NODE, 6-3
 EDITOR, 6-4
 PERSONAL_MESSAGE, 6-4

MAIL

initialization file parameters (Cont.)

REPLY_ADDRESS, 6-2

qualifiers

/EDIT, 6-8

/SUBJECT, 6-8

using, 6-8

starting, 6-5

using distribution lists, 6-6

N

NDU, 5-1

command format, 5-7

commands

CLOSE, 5-13

CREATE, 5-15

DELETE, 5-17

EXIT, 5-6

HELP, 5-11, 5-18

OPEN, 5-20

SHOW, 5-22

command summary

remote printers, 5-8

virtual disks, 5-8

ending connection with remote disk,
5-13

error messages

see Appendix B

exiting from, 5-6

getting help for, 5-11, 5-18

prompts, 5-6

sample disk status display, 5-22

sample printer status display, 5-22

starting (virtual disks), 5-5 to 5-6

Network Device Utility

see NDU

Network File Transfer utility

see NFT

Network Mail Utility

see MAIL

NFT, 4-1, E-1

abbreviating commands, 4-4

abbreviating switches, 4-4

access-control information, 4-7

command format, 4-3

commands

APPEND, 4-14, 4-20

COPY, 4-11, 4-14, 4-23

DELETE, 4-15, 4-29

DIRECTORY, 4-16, 4-30

entering, 4-3

EXIT, 4-33

HELP, 4-4, 4-5, 4-34

PRINT, 4-36

SET, 4-7, 4-9, 4-37

SHOW, 4-10, 4-38

SUBMIT, 4-39

TYPE, 4-17, 4-40

command summary, 4-19

error messages

see Appendix A

exiting from, 4-3, 4-33

file operations, 4-13

help, 4-5, 4-34

starting, 4-2 to 4-3

from command line, 4-2

from windows, E-1

switches

/BRIEF, 4-16

for APPEND, 4-20

for COPY, 4-24

for DELETE, 4-29

for DIRECTORY, 4-30

for HELP, 4-34

for PRINT, 4-36

for SUBMIT, 4-39

/IMAGE, 4-12, 4-13

/NOCONVERT, 4-12

/PRINT, 4-16

summary of, 4-4

using switches, 4-4

Nonstream file system, 4-12

P

Password, 4-6

PORT, F-8

Printing files, 4-16, 4-36

Prompts

NDU, 5-6

NFT, 4-4

R

Remote files

accessing, 4-1, A-1

copying to a local node, 4-12, 4-15

printing, 4-16

Remote printers, 5-5

closing down connection, 5-13

creating a print queue, 5-15

S

Server functions

using the Job Spawner

with DTR, 2-1, 2-4

with FAL, 2-1

Stream file system, 4-11

Switches

NFT, 4-4

Systems

types of, 4-11

U

User name, 4-6

V

Virtual disks

creating, 5-15

deleting space, 5-17

establishing connection with virtual
disk, 5-20

volumes, 5-4

W

Wildcards

asterisks, 4-16, A-2

combining, A-2

in file specifications, A-2

percent signs, A-2

question marks, A-2

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
call 800-DIGITAL

In Canada
call 800-267-6215

In New Hampshire
Alaska or Hawaii
call 603-884-6660

In Puerto Rico
call 809-754-7575

ELECTRONIC ORDERS (U.S. ONLY)

Dial 800-DEC-DEMO with any VT100 or VT200
compatible terminal and a 1200 baud modem.
If you need assistance, call 1-800-DIGITAL.

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

INTERNATIONAL

DIGITAL
EQUIPMENT CORPORATION
A&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC),
Digital Equipment Corporation, Westminister, Massachusetts 01473

* Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809-754-7575 x2012

READER'S COMMENTS

What do you think of this manual? Your comments and suggestions will help us to improve the quality and usefulness of our publications.

Please rate this manual:

	Poor			Excellent	
Accuracy	1	2	3	4	5
Readability	1	2	3	4	5
Examples	1	2	3	4	5
Organization	1	2	3	4	5
Completeness	1	2	3	4	5

Did you find errors in this manual? If so, please specify the error(s) and page number(s).

General comments:

Suggestions for improvement:

Name _____ Date _____

Title _____ Department _____

Company _____ Street _____

City _____ State/Country _____ Zip Code _____

DO NOT CUT - FOLD HERE AND TAPE



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY LABEL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

digital™

**Networks and
Communications Publications**
550 King Street
Littleton, MA 01460-1289



DO NOT CUT - FOLD HERE

digital